NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ANALYSIS, DESIGN AND IMPLEMENTATION OF A PROOF-OF-CONCEPT PROTOTYPE TO SUPPORT LARGE-SCALE MILITARY EXPERIMENTATION

by
Eid S. Alqhatani
Fahad A. Bin Huwaymil

September 2013

Thesis Co-Advisors:	Magdi N. Kamel
	Walter Kendall
Second Reader:	Douglas Brinkley

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE ANALYSIS, DESIGN AND IMPLEMENTATION OF A PROOF-OF-CONCEPT PROTOTYPE TO SUPPORT LARGE-SCALE MILITARY EXPERIMENTATION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Eid S. Alqhatani and Fahad A. Bin Huwaymil	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT

FORCEnet Innovation and Research Enterprise (FIRE) is an enterprise computer-based solution developed to support large-scale experimentation in the Navy and Department of Defense. Every year, experiments are conducted such as Trident Warrior (TW) events to assess new capabilities developed to achieve FORCEnet concept. FIRE is also used to support experimentation in other projects and for other services. FIRE was built by the Naval Postgraduate School to provide the necessary tools for the coordination of the planning, execution and reporting of these experiments. Since its inception in 2003, FIRE has played an essential role in TW by empowering all stakeholders with the collaborative and management tools to perform tasks that were time-consuming and manpower-intensive in the past. However, a survey conducted a few years ago showed that FIRE lacks some required features and improvement in various areas needed to be considered. The objective of this thesis was to design, develop, and test a proof-of-concept prototype of an improved web-based application to support the coordination of large-scale experimentation to address the shortcomings of the old FIRE system. This was achieved by using the following: a modern design approach; the Model-View-Controller; a state-of-the-art framework; Oracle Application Development Framework; and powerful development tools such as Oracle JDeveloper and Oracle WebCenter.

14. SUBJECT TERMS: FORCEnet, FIRE, Web-based application, Oracle ADF, Trident Warrior, Experiment, MVC	**15. NUMBER OF PAGES** 119
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

ANALYSIS, DESIGN AND IMPLEMENTATION OF A PROOF-OF-CONCEPT PROTOTYPE TO SUPPORT LARGE-SCALE MILITARY EXPERIMENTATION

Eid S. Alqahtani
Lieutenant Commander, Royal Saudi Naval Forces
M.S., École Polytechnique, 2004

Fahad A. Bin Huwaymil
Lieutenant Commander, Royal Saudi Naval Forces
M.S., École Polytechnique, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT
and
MASTER OF BUSINESS ADMINISTRATION

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Authors:	Eid S. Alqahtani Fahad A. Bin Huwaymil
Approved by:	Magdi Kamel Thesis Co-Advisor
	Walter Kendall Thesis Co-Advisor
	Douglas Brinkley Second Reader
	Dan C. Boger Chair, Department of Information Science
	William Gates Dean, Graduate School of Business and Public Policy

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

FORCEnet Innovation and Research Enterprise (FIRE) is an enterprise computer-based solution developed to support large-scale experimentation in the Navy and Department of Defense. Every year, experiments are conducted such as Trident Warrior (TW) events to assess new capabilities developed to achieve FORCEnet concept. FIRE is also used to support experimentation in other projects and for other services. FIRE was built by the Naval Postgraduate School to provide the necessary tools for the coordination of the planning, execution and reporting of these experiments. Since its inception in 2003, FIRE has played an essential role in TW by empowering all stakeholders with the collaborative and management tools to perform tasks that were time-consuming and manpower-intensive in the past. However, a survey conducted a few years ago showed that FIRE lacks some required features and improvement in various areas needed to be considered. The objective of this thesis was to design, develop, and test a proof-of-concept prototype of an improved web-based application to support the coordination of large-scale experimentation to address the shortcomings of the old FIRE system. This was achieved by using the following: a modern design approach; the Model-View-Controller; a state-of-the-art framework; Oracle Application Development Framework; and powerful development tools such as Oracle JDeveloper and Oracle WebCenter.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADF	Application Development Framework
AM	Application Module
BC	Business Components
C4ISR	Command and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
C2	Command and Control
CRUD	Create, Read, Update and Delete
COI	Critical Operational Issue
CRM	Customer Relation Management
DDL	Data Definition Language
DBMS	Database Management System
DoD	Department of Defense
DON	Department of the Navy
ECs	Enabling Capabilities
EJB	Enterprise Java Beans
EO	Entity Object
ERD	Entity-Relationship Diagram
XE	Express Edition
XML	eXtensible Markup Language
FIRE	FORCEnet Innovation and Research Enterprise
FNC	Future Naval Capabilities
HR	Human Resources
HTML	HyperText Markup Language
IA	Information Assurance
IDE	Integrated Development Environment
IPT	Integrated Product Team
JCR	Java Content Repository
Java EE	Java Enterprise Edition
JSF	Java Server Faces
JVM	Java Virtual Machine

JSR170	Java Specification Request 170
JSP	JavaServer Pages
JP	Joint Publication
KM	Knowledge Management
LOVs	List of Values
MEL	Master Event List
MSEL	Master Scenario Event List
MTBF	Mean Time Between Failure
MDS	MetaData Services
MVC	Model-View-Controller
NETWARCOM	Naval Network Warfare Command
NPS	Naval Postgraduate School
ONR	Office of Naval Research
OCS	Oracle Collaboration Suite
POC	Point of Contact
PM	Program Manager
RDT&E	Research, Development, Test and Evaluation
S&T	Science and Technology
SOA	Service-Oriented Architecture
SPAWAR	Space and Naval Warfare
TTPs	Tactics, Techniques & Procedures
TOG	Technology Oversight working Group
T&E	Test and Evaluation
TBF	Time Between Failure
TW	Trident Warrior
UML	Unified Modeling Language
URL	Uniform Resource Locator
UCM	Universal Content Management
UI	User Interface
VO	View Object
WYSIWYG	what you see is what you get

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Professor Magdi Kamel for his patience, motivation, enthusiasm, and knowledge. The guidance and direction he has provided over the last several quarters helped us research and write this thesis. We could not have imagined having a better advisor and mentor for our research. In addition, we would also like to express our deepest gratitude to Mr. Walter Kendall for the valuable knowledge and insights he has provided to us. The training and advice you provided during the research proved invaluable. Also, we would like to thank Dr. Douglas Brinkley for taking the time to be our second reader.

Eid expresses his love and gratitude to his wife, Sirine, and his sons, Saad and Youssef:

> Sirine, your support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which my life have been built. You gave me unconditional support and love through all this long process. Saad and Youssef, you are the source of my happiness.
>
> Fahad thanks his family for their love and never-ending support.
>
> Without your encouragement and understanding it would have been impossible for me to finish this work

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

In 2002, the U.S. Department of the Navy (DON) initiated the Future Naval Capabilities (FNC) program, a Science and Technology (S&T) program. The goal of FNC is to address the capability gaps of the Navy and Marine Corps through technology investments called enabling capabilities (ECs) [1]. Each EC consists of one or more technological product(s) intended to close existing capability gaps. FNC products fall into nine functional areas of development, called pillars. FORCEnet is one of these pillars and is considered the architectural framework for naval warfare in the Information Age. Each year, ECs in areas such as Command and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR), networking, navigation, decision support and space technologies are examined, and new capabilities are added to implement the FORCEnet concept. The Trident Warrior (TW) is the Navy's major FORCEnet Sea Trial event. TW is the experimental platform for testing the ECs of the FORCEnet integration efforts. The participants in TW need a collaborative system to track experiments coordinate tasks, plan events and share data in order to achieve the mission effectively and efficiently. Therefore, the Knowledge Management (KM) Lab at the Naval Postgraduate School (NPS) has developed a system called FORCEnet Innovation & Research Enterprise (FIRE) to support experimentation, and it was first used in the Trident Warrior (TW) experiments in 2003.

In 2003, a survey was conducted to evaluate the value added by using FIRE in the TW planning and analysis process [2]. FIRE users were asked to state what improvements or features need to be included to the FIRE system. Their answers involved the following requirements [2]:

- Better organization of information
- One screen/page to minimize scrolling

- Ability to enter unlimited rich text and graphics into planning and data reduction areas
- Ability to format text without having to code in HTML
- Ability to post and manage own documents
- Faster response by the website
- General availability of the collaboration services

Moreover, many new features have become available due to advancements in software technology since 2003.

Using the Oracle Application Development Framework (ADF), this research aims to design and implement a proof-of-concept web-based application to support the coordination of large-scale experimentation. We call the resulting application *i*FIRE because it represents a significant improvement to the actual FIRE system.

B. OBJECTIVE

The purpose of this research is to design and implement a proof-of-concept web-based application to support the planning, execution, analysis, reporting, and coordination of large-scale experimentation; therefore, it will facilitate the testing of FORCEnet components.

C. PROBLEM STATEMENT

Integration of all warriors, sensors, networks, command and control systems, platforms and weapons into a networked, distributed combat force can be achieved through extensive experimentation. The nature of this experimentation is incremental in nature. In order to coordinate all aspects of an experiment effectively, there is a requirement for a system to serve as a single point of information where any and all experiment participants can go to develop test plans, coordinate activities, form teams, exchange views, share documents and prepare reports.

D. SCOPE AND METHODOLOGY

1. Scope

As our research is basically focused on the design and development of a proof-of-concept web application, we will not require any specific data to complete our research analysis. The application will be useful for the planning, execution, analysis, reporting, and coordination of the ongoing experimentation for FORCEnet concept implementation. However, we foresee that a complete solution (inception to end use) may not be possible due to the limited available time.

2. Methodology

The methodology used for this research will consist of the following steps:

- Perform a literature review and evaluation of the current system;
- Complete a requirement analysis and validation;
- Learn the Oracle tools used for application development and discover the capabilities that could be added by these tools;
- Use a rapid prototyping approach for application development to allow early feedback from users and improve the prototype;
- Test and evaluate the prototype;
- Capture the shortcomings/missing functionalities required by users;
- Revise the prototype to address shortcomings identified by users; and if time permits,
- Re-test and improve the application.

3. Primary Research Question

Can we build a web-based application using the model-view-controller (MVC) methodology as implemented in the Oracle Application Development Framework (ADF) to support the planning, execution, analysis, reporting and coordination of large-scale experimentation?

4. Benefits of Study

The success of this web-based application prototype is the first step of a FIRE system upgrade/improvement. Findings of this research will serve is a solid foundation for future studies aiming to improve FIRE. Consequently, this effort would be very beneficial for the planning, execution, analysis, reporting as well as coordination of large-scale experimentation and would contribute to its long term success.

E. ORGANIZATION OF THESIS

This thesis consists of seven chapters:

- **Chapter I: Introduction.** This chapter gives a general outline of the problem with a description of the research scope and methodology, and the organization of the thesis.
- **Chapter II: FORCEnet, Trident Warrior and FIRE**. This chapter discusses the background of the Future Naval Capabilities program, the FORCEnet concept and the evolution of Trident Warrior. It will also discuss the FORCEnet Innovations and Research Enterprise system.
- **Chapter III: Requirements Analysis.** This chapter discusses the application requirement analysis and defines the data model and use cases.
- **Chapter IV: Development Approach and Tools**. This chapter discusses the development approach used in this research, the Oracle Application Development Framework and the tools used for development, including JDeveloper, SQL Developer and WebCenter.
- **Chapter V: Prototype Implementation.** This chapter describes the elements of the system as well as the application features and tools available to the users.
- **Chapter VI: Test and Evaluation**. This chapter discusses the system test methodology and analyzes the users' feedback.
- **Chapter VII: Conclusion**. This chapter summarizes the key findings and conclusions drawn from this thesis, and offers recommendations for future research in this area.

II. FORCENET, TRIDENT WARRIOR AND FIRE

A. FUTURE NAVAL CAPABILITIES

In 2002, the Department of the Navy initiated the Future Naval Capabilities (FNC) program to address the capability gaps of the Navy and Marine Corps through technology investments called enabling capabilities (ECs) [1]. Each EC consists of one or more technological product(s) intended to close identified capability gaps. FNC products fall into nine functional areas of development (pillars): capable manpower, enterprise and platform enablers, expeditionary maneuver warfare, force health protection, FORCEnet, power and energy, sea basing, sea shield and sea strike [1].

The FNC program is designed to develop and transition cutting-edge technology products to acquisition managers within a three- to five-year timeframe. The program objective is to deliver mature products to warfighters to enhance their warfighting and support capabilities. Each FNC pillar is managed by a dedicated integrated product team (IPT) and IPT working groups headed by a two-star flag officer [1].

All stakeholders are involved in the program's oversight, management and execution throughout the product development life cycle. The program is driven by the requirements from field commands. These requirements are translated into functional capabilities, and analysis of alternatives is carried out by the office of naval research (ONR) for all possible solutions [1]. The best solution is assigned to one of the nine pillars of FNC as an EC and approved by the technology oversight working group (TOG) for resource allocation for further acquisition.

The time required for any EC to mature from concept to a tested product varies between three to five years. Once the technology is demonstrated, a product is formally brought into the programs of record and is programmed for induction into service through the acquisition community. The Program Manager

(PM) takes further responsibility for conducting any additional research, development, test and evaluation (RDT&E) necessary to engineer and integrate the product into an EC. Figure 1 depicts the life cycle process of each EC from requirement generation to resource allocation.

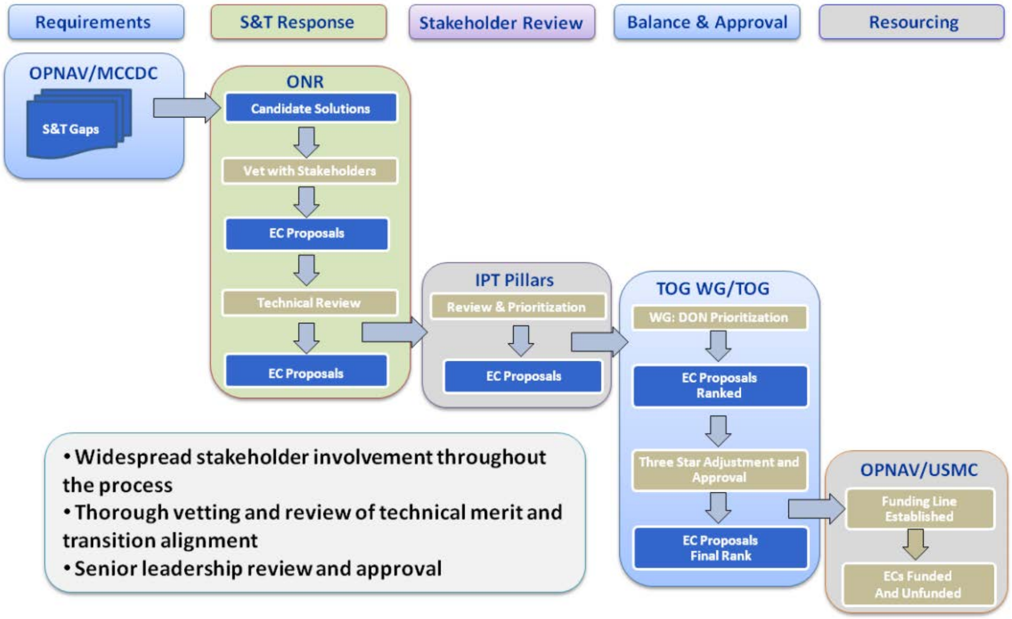


Figure 1. Life cycle process for an EC from requirement generation to resource allocation. From [1].

B. FORCENET CONCEPT

1. Description

The modern battlefield has become exceedingly complex and technology driven. Informed decision making at levels of force structure is the call of the day. Availability of high mobility, long range accurate, and lethal fire power, coupled with sophisticated surveillance systems has made it possible, at least in theory, to enable all involved in action to have real-time access to information related to ever changing battlefield situations. The operational focus shift from conventional warfare to asymmetric warfare demands even more accuracy and precision in the use of kinetic weapons to avoid any unwanted collateral damage.

The FORCEnet concept is meant to empower sailors and marines at all levels to execute more effective decision making at an increased tempo. FORCEnet is the naval command and control component of Sea Power 21 (Figure 2) and Expeditionary Warfare. FORCEnet can be defined as "the operational construct and architectural framework for naval warfare in the Information Age, integrating warriors, sensors, command and control, platforms, and weapons into a networked, distributed combat force" [3]. In simple terms, FORCEnet is meant for "connecting everything to everything" [3].

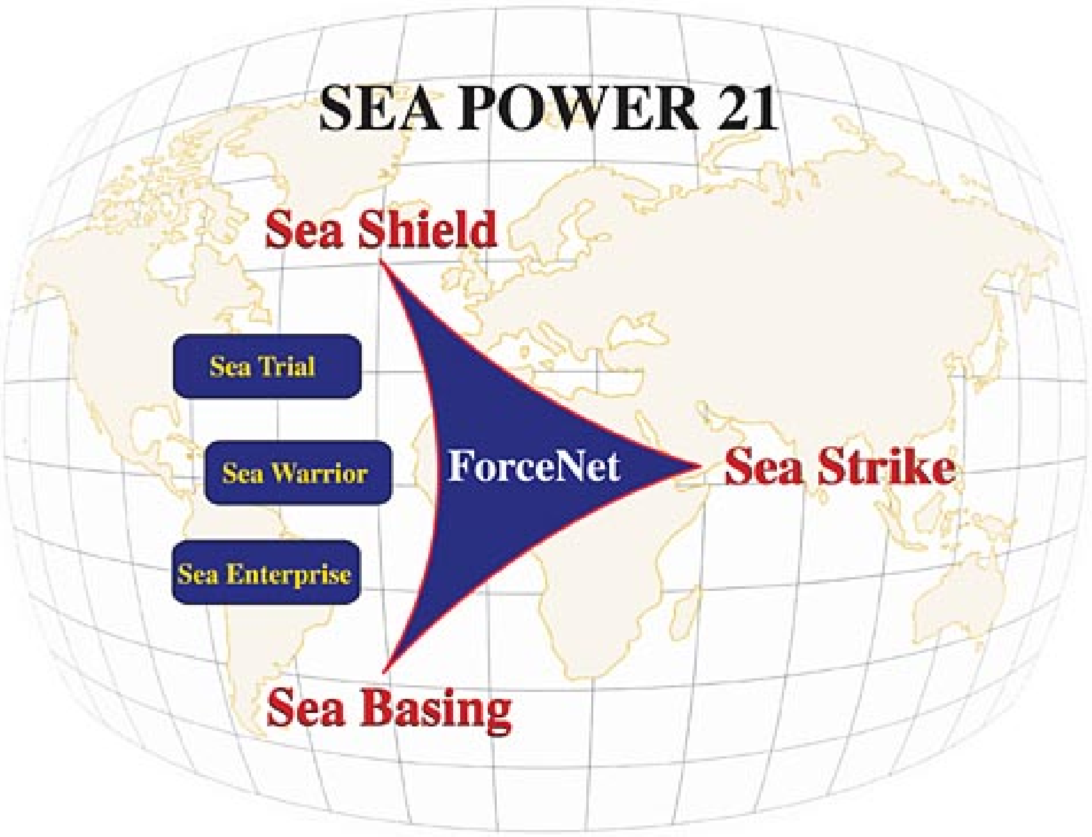


Figure 2. Overview of Sea Power 21. From [4].

FORCEnet is not an acquisition program; rather it is a roadmap for the Navy and Marine Corps future force command and control development effort.

The concept describes the power and intent of networking all assets in globally distributed networks to increase warfighting capabilities and improve overall combat effectiveness [3].

The FORCEnet concept may never reach a final state as DoD will be forced to adopt “transformational” technologies while maintaining backward integration with already in service “legacy” systems. FORCEnet is based on the hypothesis [3]:

> When all forces and organizations down to the level of individuals are interconnected in a networked, collaborative command and control environment, then all operations and activities will enjoy the benefits of decentralization….and commanders will make and implement better decisions faster than any enemy can endure.

Therefore, FORCEnet requires a continuous testing of technologies coming online with the passage of time. However, it is neither feasible nor possible to test everything at one time. Figure 3 provides a graphic representation of the FORCEnet capabilities development concept.

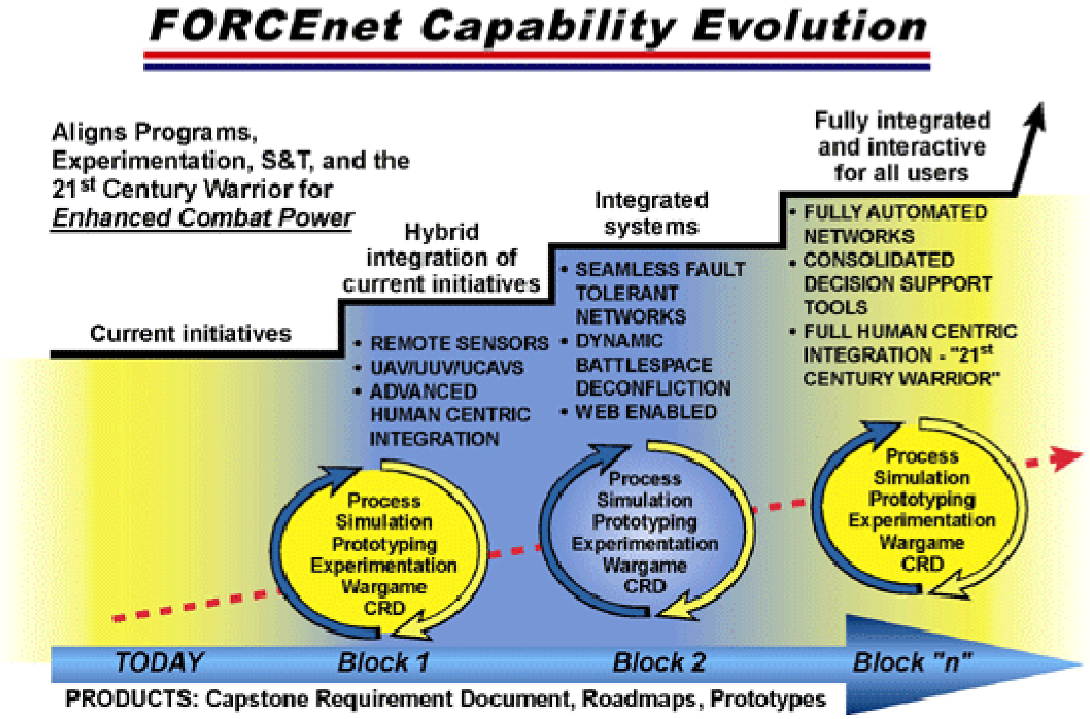

FORCEnet Capability Evolution
Aligns Programs, Experimentation, S&T, and the 21st Century Warrior for Enhanced Combat Power
Current initiatives
Hybrid integration of current initiatives
• REMOTE SENSORS
• UAV/UUV/UCAVS
• ADVANCED HUMAN CENTRIC INTEGRATION
Integrated systems
• SEAMLESS FAULT TOLERANT NETWORKS
• DYNAMIC BATTLESPACE DECONFLICTION
• WEB ENABLED
Fully integrated and interactive for all users
• FULLY AUTOMATED NETWORKS
• CONSOLIDATED DECISION SUPPORT TOOLS
• FULL HUMAN CENTRIC INTEGRATION - "21st CENTURY WARRIOR"
Process Simulation Prototyping Experimentation Wargame CRD
Process Simulation Prototyping Experimentation Wargame CRD
Process Simulation Prototyping Experimentation Wargame CRD
TODAY
Block 1
Block 2
Block "n"
PRODUCTS: Capstone Requirement Document, Roadmaps, Prototypes

Figure 3. FORCEnet capabilities development concept. From [5].

2. Capabilities

The FORCEnet Functional Concept identifies 15 core capabilities essential for FORCEnet implementation. These capabilities will, in fact, drive supporting architectures, standards and metrics which will guide subsequent programmatic requirements during implementation phase. The FORCEnet Functional Concept, once fully implemented in the Navy and Marine Corps, will provide unprecedented situational awareness, firepower and seamless alignment with joint services and coalition forces. The capabilities necessary to implement FORCEnet concept, which are described in detail in [3] and include the following [3]:

- Provide robust, reliable communication to all nodes, based on the varying information requirements and capabilities of those nodes.
- Provide reliable, accurate and timely location, identity and status information on all friendly forces, units, activities and entities.
- Provide reliable, accurate and timely location, identification, tracking and engagement information on environmental, neutral and hostile elements, activities, events, sites, platforms and individuals.
- Store, catalogue and retrieve all information produced by any node on the network in a comprehensive, standard repository so that the information is readily accessible to all nodes and compatible with the forms required by any nodes, within security restrictions.
- Process, sort, analyze, evaluate and synthesize large amounts of disparate information while still providing direct access to raw data as required.

Information is valuable only if it assists the commanders in analyzing the operational environment. This capability, therefore, requires that information that is made available in a shared space must conform to a format so that it adds value to the decision making process. Information can become more valuable when formatted into a more useful form, combined or compared with other information, and analyzed and evaluated for meaning and implications.

- Provide each decision maker the ability to depict situational information in a tailorable, user-defined, shareable, primarily visual representation.
- Provide distributed groups of decision makers the ability to cooperate in the performance of common command and control activities by means of a collaborative work environment.

- Automate lower-order command and control sub-processes and use intelligent agents and automated decision aids to assist people in performing higher-order sub-processes, such as gaining situational awareness and devising concepts of operations.
- Provide information assurance (IA).

As per joint publication, JP 3-13.1, information assurance is “concerned with measures that protect and defend information and information systems, and many of the measures involve the use of the EMS” [6]. Therefore, protecting and defending information and information systems is a vital part of FORCEnet concept.

- Function in multiple security domains and multiple security levels within a domain, and manage access dynamically.
- Interoperate with command and control systems of very different type and level of sophistication.
- Allow individual nodes to function while temporarily disconnected from the network.
- Automatically and adaptively monitor and manage the functioning of the command and control system to ensure effective and efficient operation and to diagnose problems and make repairs as needed.
- Incorporate new capabilities into the system quickly without causing undue disruption to the performance of the system.
- Provide decision makers the ability to make and implement good decisions quickly under conditions of uncertainty, friction, time, pressure, and other stresses.

C. TRIDENT WARRIOR

The Trident Warrior (TW) is an annual testing event conducted to prove newly developed capabilities, communications, networks, technologies and Tactics Techniques & Procedures (TTPs). TW is the Navy’s major FORCEnet Sea Trial event. TW is TW the experimental platform for testing the ECs of the FORCEnet integration efforts, and it is cosponsored by Naval Network Warfare Command (NETWARCOM) and Space and Naval Warfare Systems Command (SPAWAR) [7].

The coordination of TW experiments was first supported by the FIRE system in 2003 [2]. These experiments have been conducted regularly since then with code names TW 04, TW 05 and so on, where the last two digits represent the year the experiment was conducted.

D. FIRE

1. Description

FORCEnet attempts to dramatically enhance how the Navy acquires, shares and capitalizes on information superiority in order to generate transformational combat effectiveness. Implementation of FORCEnet requires large-scale military experimentation involving various legacy systems and networks connected together. FIRE was developed to facilitate experimentation planning, execution, analysis processes and reporting of these systems. FIRE was designed at NPS to support experiments by providing enterprise-level features such as document repository and information sharing. FIRE also provides collaboration tools such as chatting, video conferencing, wikis and others to coordinate the experiment’s various activities.

2. Evolution

As highlighted earlier, FIRE was introduced in 2003 to support TW03. The major purpose of FIRE at that time was as a central document repository to keep all stakeholders updated with respect to most current versions of experiment-related documentation [2]. FIRE, therefore, provided a medium for uploading documents, test procedures, test results and test objectives without the exchange of lengthy emails. The initiative development taxonomy was introduced in TW03 and served as a common structure for the initiative leads to describe their experimental objectives. FIRE helped experiment analysts and, as a result, data collection planning improved considerably as compared to the past [2].

FIRE has improved continuously over the subsequent years. Every year new features were incorporated based on past experiences and new test requirements. Features like the detailed data planning area and Master Scenario Event List (MSEL) were added in TW 04 [2]. The MSEL served as an event planner. Experiment personnel have the flexibility of sorting MSEL by platform, date, time or initiative. This feature helps experiment personnel to organize data collectors, data collection and data execution and to assign the right people, at the right time and at the right place to collect the data needed to support the objective at stake.

The data planning taxonomy was improved, and MSEL was replaced with the Master Event List (MEL) in TW 05 which was a "much simpler format than its scenario-based predecessor" [2]. The Oracle Collaboration Suite (OCS) was also integrated into FIRE during 2005 experiments in TW05. The OCS included a Web Conferencing tool that greatly improved communication and facilitated enhanced collaboration between the various key experiment leads and planners [2].

This research aims to contribute to the evolution of FIRE. A successful development of a proof-of-concept prototype will mark the start of developing others prototypes. Consequently, those prototypes will lead to a new generation of FIRE application and service with more capabilities and features.

E. SUMMARY

In this chapter, we described FORCEnet concept and TW and discussed the need for adding new capabilities, how to plan for them and test them. We also described the actual FIRE system and its evolution and the role it plays for the coordination of large-scale experimentation. In Chapter III, we discuss the requirements analysis, which is the first step in the development of *i*FIRE.

III. SYSTEM REQUIREMENT ANALYSIS

This chapter will focus on identifying the *i*FIRE system requirements. The requirement analysis is the cornerstone of any information system design; therefore, an effort was made to make sure that all requirements were included during the requirement analysis phase. We conducted our requirement analysis by analyzing the actual FIRE system functionalities and its experiment planning and reporting structures, interviewing some subject matter experts in the KM lab at NPS and reviewing the users' feedback produced in a previous study. The results of our requirements analysis give us a firm understanding of what the users need and the tasks the system must perform.

The results of the requirements analysis are broken into the following categories: data business requirements, process business requirements, workspace requirements, and security requirements. The data business requirements analysis allowed us to build the logical data model of the system. The process business requirements analysis resulted in the definition of the use cases which represent the activities the *i*FIRE system is designed to perform. The workspace requirements analysis allowed us to specify all the collaboration tools required for the users to achieve their tasks. The security requirements analysis allowed us to define the security level and the users' access privileges.

In the remainder of this chapter, we define the purpose of the system. Then, we discuss the data business requirement throughout the description of the experiment coordination phases, which is followed by the definition of the data model. Following that, we discuss the process business requirement throughout the identification of the activity the system is designed to perform. Finally, the last three sections of this chapter provide a description of the workspace, security and hardware and software requirements.

A. PURPOSE OF THE SYSTEM

The purpose of the system is to coordinate and track the planning, execution and reporting of large-scale experimentation.

B. DATA BUSINESS REQUIREMENT

1. Experiment Coordination Phases

According to the experiment planning and reporting structures, the coordination of an experiment involves the following five phases: Objective Planning, Data Planning, Event Management, Result Reporting and Result Analysis. We will discuss the requirements of each phase in some detail.

a. Objective Planning

In this phase, the objective of an experiment/test is defined within a specific focus area (e.g.,, Electronic Warfare, Command and Control, etc.). The goal of defining this objective is to respond to a specific Critical Operational Issue (COI). An objective is associated with one or many objective-questions that specify what is to be learned. Answering an objective-question may require one or all of the following: system, human involvement and surveys. Also, for each objective-question, one or more measures will need to be specified. These measures will be achieved through the use of a specific System and Method [8]. Table 1 shows an example of the objective planning elements.

Focus Area	Command and Control
COI	A network is needed that enables information communication across all battlefield units.
Objective	Provide the capability to interoperate across entities.
Objective-Question	Can the system provide reliable connectivity? Can the system provide persistent connectivity?
Measure	Time Between Failure (TBF) Mean Time Between Failure (MTBF)

Table 1. Example of objective planning main elements.

For each objective, a point of contact (POC) is designated. The POC leads the objective analysis task and manages any objective or objective-question modification requests.

An objective-question with its set of measures, method, system and data needed to answer that objective-question, defines an experiment thread. Each thread is given a code which is the key to be used for archiving and retrieval of information to/from the database. The thread code consists of the focus area, the COI code, the objective and the objective-question numbers. Figure 4 depicts the thread numbering code.

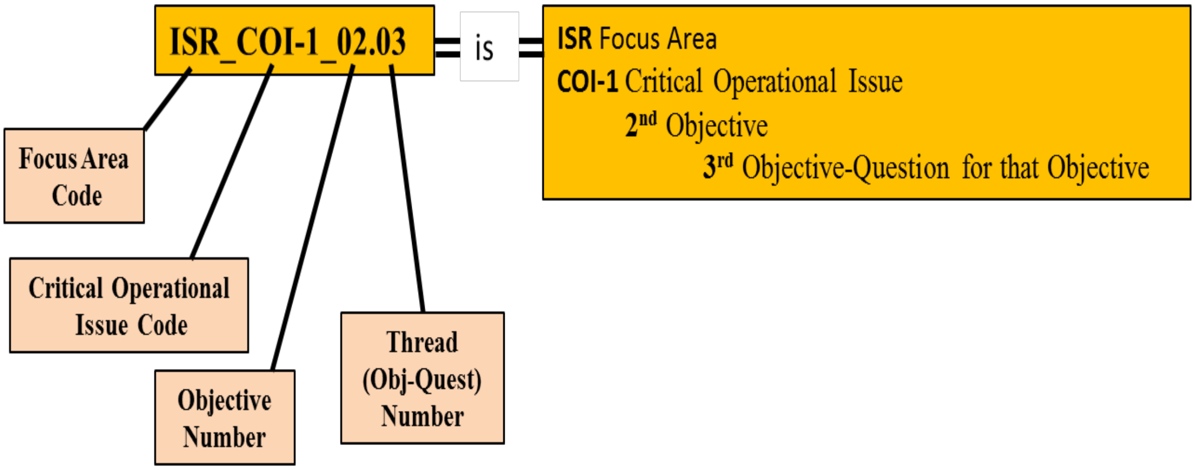

ISR_COI-1_02.03
is
ISR Focus Area
COI-1 Critical Operational Issue
2nd Objective
3rd Objective-Question for that Objective
Focus Area Code
Critical Operational Issue Code
Objective Number
Thread (Obj-Quest) Number

Figure 4. Thread number code. After [8].

b. Data Planning

In this phase, the users specify the data to be collected for each required measure that was defined in the objective planning phase. The data to be collected during the experiment fall into one of the following categories:

- Sniffer data
- Ground truth data
- System-derived data
- Observation data
- Survey data
- Interview data

c. Event Management

In this phase, users must specify the experiment events in which the required data for each objective-question will be produced. An event could be a technical test, survey or others. Each event may be dedicated for one or more objective-question.

Users should specify the following:

- Event summary
- Event date
- Location
- Operation condition
- System condition
- Information condition
- Operator requirement.

After the execution of each event, users should report any deviation between the event planning and the event execution if there is any.

d. Result Reporting

After the experiment event execution, results are collected and reported. Each set of results is linked to a specific measure that is required by one objective-question. At the end of this phase, results are ready for analysis and validation.

e. Result Analysis

In this phase, users and analysts check all the results per objective-question. Then, they consolidate all these results to form one result for each objective-question, and thus, they form one result for their respective objective. Finally, they assess and report the validity of these results. That is, they confirm whether the objective-questions were answered and the objective is achieved.

2. Data Model

a. Entities Definition

In sub-section B.1, a description of the experiment coordination phases was provided. In this section, we identify the “things” the users want to track in each phase. These “things” are the data entities the system will store as their information.

Table 2 shows, by phase, all the entities to be implemented for the data model of the *i*FIRE system.

Phase	Entities
Objective Planning	• Focus Area • COI • Objective • Objective Analysis • POC • Objective-Question • Measure • System • Method
Data Planning	• Data
Event Management	• Event • Executed Event
Result Reporting	• Result
Result Analysis	• Objective Result • Objective Question Result

Table 2. Data Model Entities per phase.

b. Entity-Relationship Diagram

Figure 5 shows the Entity-Relationship Diagram (ERD) of the *i*FIRE system's 15 entities. The diagram shows the entities relationship as well as the minimum and maximum cardinality constraints.

The *i*FIRE system data dictionary, provided in Appendix A, shows the details of the entity attributes, their types and size. It also shows the primary keys (identifiers) and foreign keys of each entity.

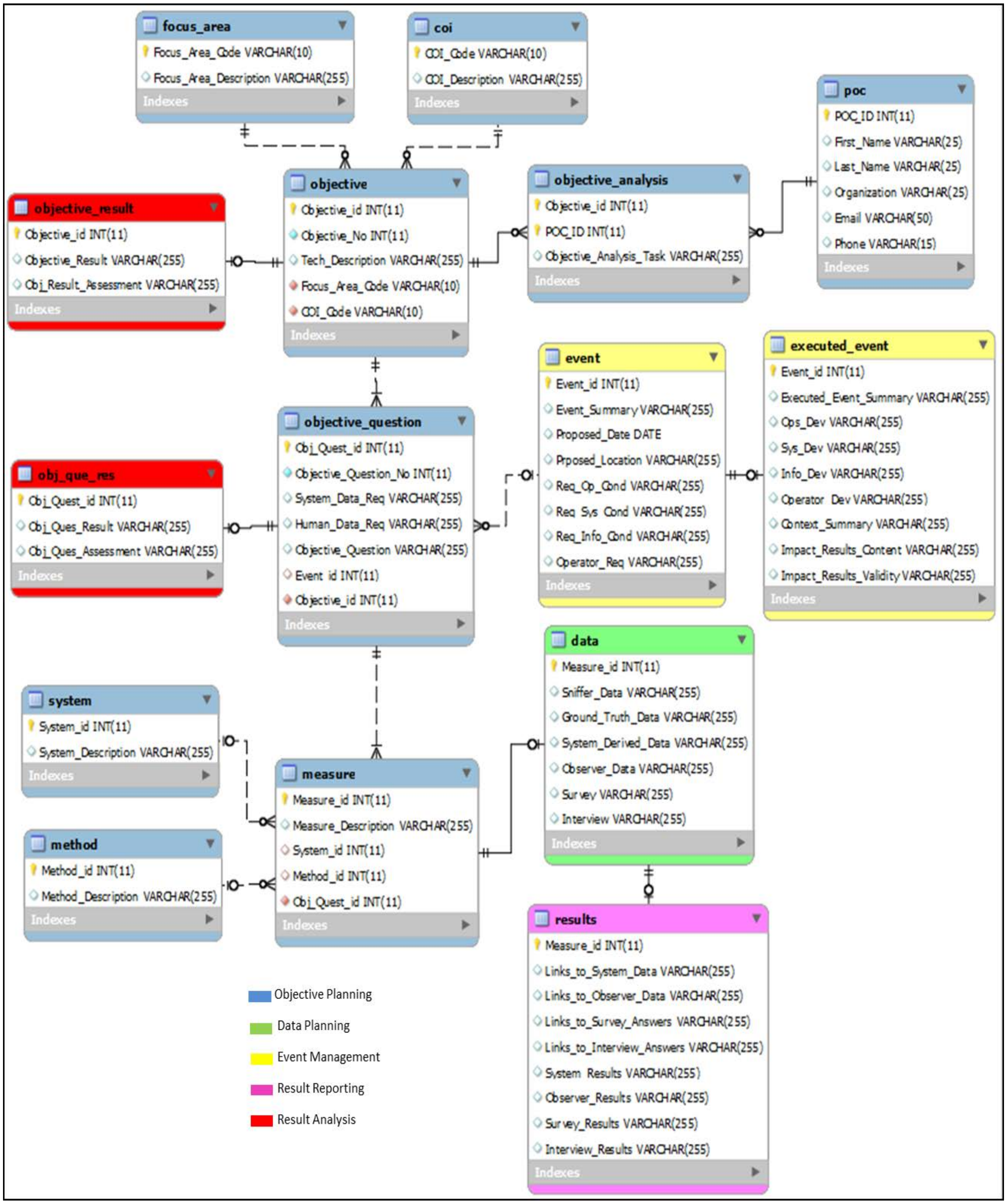

focus_area
Focus_Area_Code VARCHAR(10)
Focus_Area_Description VARCHAR(255)
Indexes
coi
COI_Code VARCHAR(10)
COI_Description VARCHAR(255)
Indexes
poc
POC_ID INT(11)
First_Name VARCHAR(25)
Last_Name VARCHAR(25)
Organization VARCHAR(25)
Email VARCHAR(50)
Phone VARCHAR(15)
Indexes
objective_result
Objective_id INT(11)
Objective_Result VARCHAR(255)
Obj_Result_Assessment VARCHAR(255)
Indexes
objective
Objective_id INT(11)
Objective_No INT(11)
Tech_Description VARCHAR(255)
Focus_Area_Code VARCHAR(10)
COI_Code VARCHAR(10)
Indexes
objective_analysis
Objective_id INT(11)
POC_ID INT(11)
Objective_Analysis_Task VARCHAR(255)
Indexes
event
Event_id INT(11)
Event_Summary VARCHAR(255)
Proposed_Date DATE
Prposed_Location VARCHAR(255)
Req_Op_Cond VARCHAR(255)
Req_Sys_Cond VARCHAR(255)
Req_Info_Cond VARCHAR(255)
Operator_Req VARCHAR(255)
Indexes
executed_event
Event_id INT(11)
Executed_Event_Summary VARCHAR(255)
Ops_Dev VARCHAR(255)
Sys_Dev VARCHAR(255)
Info_Dev VARCHAR(255)
Operator_Dev VARCHAR(255)
Context_Summary VARCHAR(255)
Impact_Results_Content VARCHAR(255)
Impact_Results_Validity VARCHAR(255)
Indexes
obj_que_res
Obj_Quest_id INT(11)
Obj_Ques_Result VARCHAR(255)
Obj_Ques_Assessment VARCHAR(255)
Indexes
objective_question
Obj_Quest_id INT(11)
Objective_Question_No INT(11)
System_Data_Req VARCHAR(255)
Human_Data_Req VARCHAR(255)
Objective_Question VARCHAR(255)
Event_id INT(11)
Objective_id INT(11)
Indexes
data
Measure_id INT(11)
Sniffer_Data VARCHAR(255)
Ground_Truth_Data VARCHAR(255)
System_Derived_Data VARCHAR(255)
Observer_Data VARCHAR(255)
Survey VARCHAR(255)
Interview VARCHAR(255)
Indexes
system
System_id INT(11)
System_Description VARCHAR(255)
Indexes
measure
Measure_id INT(11)
Measure_Description VARCHAR(255)
System_id INT(11)
Method_id INT(11)
Obj_Quest_id INT(11)
Indexes
method
Method_id INT(11)
Method_Description VARCHAR(255)
Indexes
results
Measure_id INT(11)
Links_to_System_Data VARCHAR(255)
Links_to_Observer_Data VARCHAR(255)
Links_to_Survey_Answers VARCHAR(255)
Links_to_Interview_Answers VARCHAR(255)
System_Results VARCHAR(255)
Observer_Results VARCHAR(255)
Survey_Results VARCHAR(255)
Interview_Results VARCHAR(255)
Indexes
🟦 Objective Planning
🟩 Data Planning
🟨 Event Management
🟪 Result Reporting
🟥 Result Analysis

Figure 5. *i*FIRE Entity Relationship Diagram.

C. PROCESS BUSINESS REQUIREMENTS

This section focuses on the system’s functional requirements—that is, the identification of the activities the system must perform. To identify these activities, which are based on the experimentation rules and procedures, we apply the use case concept. Use cases represent the activities the system is designed to perform. Usually, the occurrence of these activities is a response to users' requests; therefore, the system stakeholders/users and their respective responsibilities are identified first.

1. *i*FIRE users

Four groups of users will be interacting with the *i*FIRE system: Guidance, Data Control, Analysis and Admin group.

a. Guidance Group

Members of this group are responsible for defining the objective (initiative) of an experiment. They specify the COI to be addressed. That is, what is the targeted operational capability throughout the achievement of objectives? They start by defining the focus areas of an experiment and then specify the COI. The objectives will be aligned into specific focus areas and COIs. For each objective, a member of this group is assigned as the POC who will lead the objective analysis task and manage any change requests. These change requests may be submitted by a member of the guidance group or a member of the data control group.

b. Data Control Group

Members of this group are responsible for defining the objective-questions for a specific objective and respectively their required measures, systems, methods and the data to be collected. They should define and manage the experiment's events information. Moreover, they are responsible for the result reporting after the experiment's completion.

c. Analysis Group

Members of this group are responsible for analyzing and assessing the objectives and objective-questions results. They review the experiment's results and provide the objective-questions and objectives results. Based on these results, they generate the objective-questions and objectives' validity. That is, they state whether the objective-questions were answered and whether their respective objective was achieved. This group could contain members of the Guidance and the Data Control groups.

d. Admin Group

Admin group members administer the *i*FIRE system. They maintain the system, deal with any issue the users may face, manage the users' accounts and manage the system access privileges. They also manage the resources available for users at run time. That means they set the level of allowable customization and personalization available at run time. Figure 6 summarizes the four group's roles.

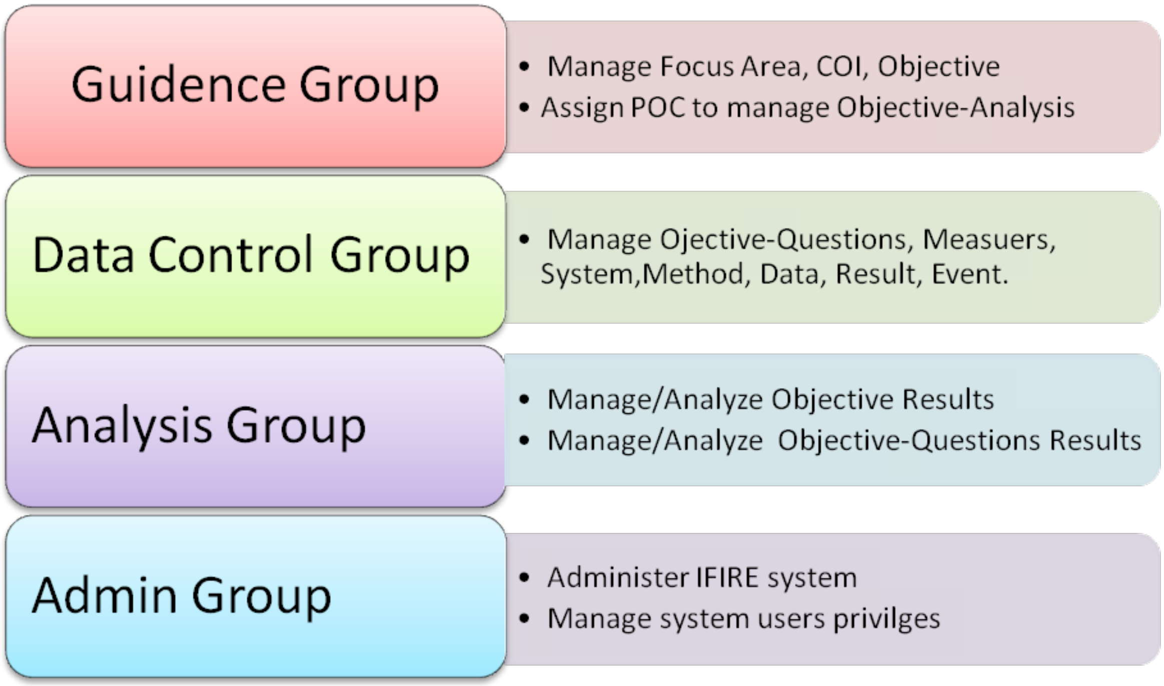
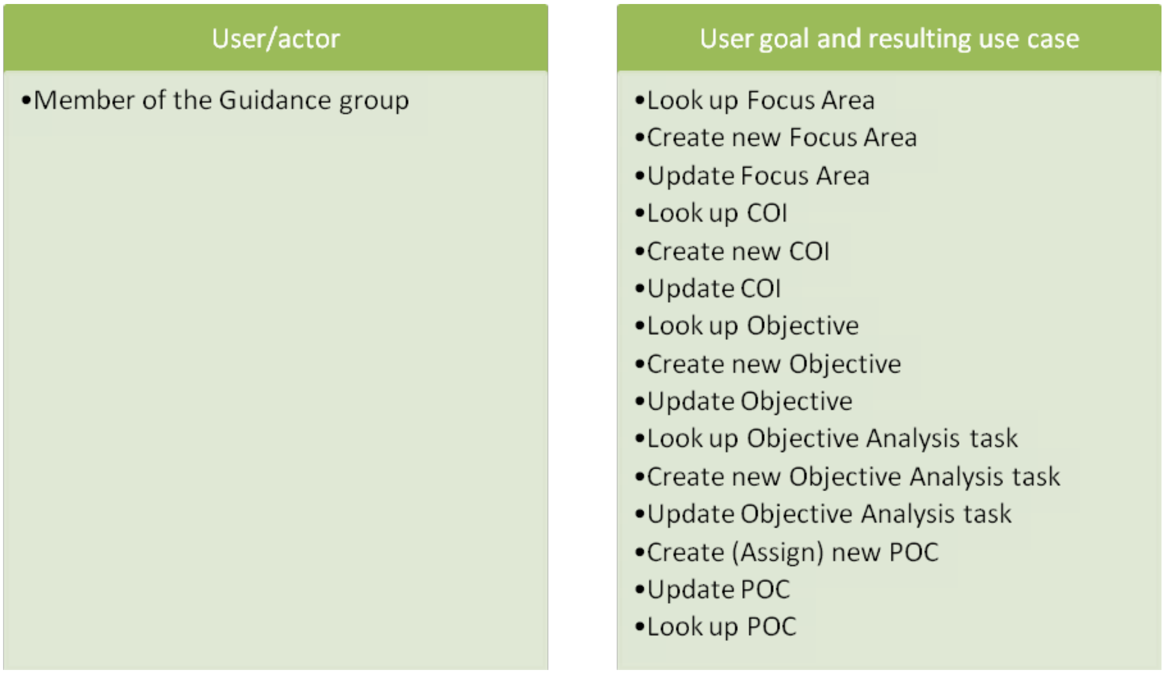

Guidence Group
Manage Focus Area, COI, Objective
Assign POC to manage Objective-Analysis
Data Control Group
Manage Ojective-Questions, Measuers, System,Method, Data, Result, Event.
Analysis Group
Manage/Analyze Objective Results
Manage/Analyze Objective-Questions Results
Admin Group
Administer IFIRE system
Manage system users privilges

Figure 6. *i*FIRE users groups' roles.

2. Use Cases

Techniques such as event decomposition technique, user goal technique and CRUD (Create, Read, Update and Delete) technique are recommended for identifying uses cases [9]. The event decomposition technique is the most comprehensive technique, and it focuses on identifying the events to which the system must respond. Events are usually triggered by the interaction of a user with the system. There are different types of events, and the sequences of events occurrence must be specified. This technique provides much detail which is beyond the size and complexity of the *i*FIRE system. In the user goal technique, all the system functions and the functionalities requested per the users are listed. Then, the user goals are established using that list. Figure 7 shows the use cases for a member of the Guidance group identified using the user goal technique.



User/actor

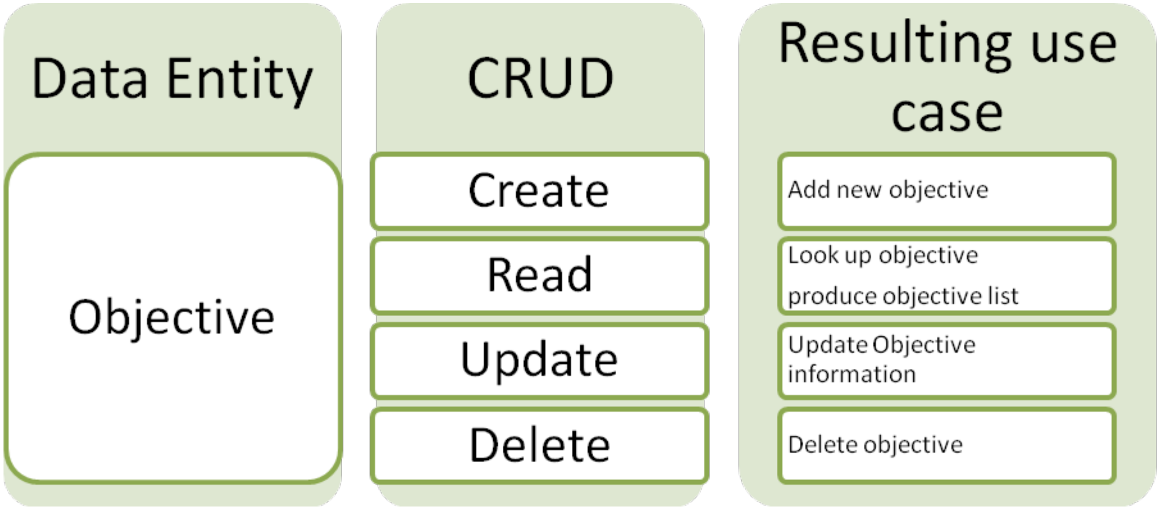
- Member of the Guidance group

User goal and resulting use case

- Look up Focus Area
- Create new Focus Area
- Update Focus Area
- Look up COI
- Create new COI
- Update COI
- Look up Objective
- Create new Objective
- Update Objective
- Look up Objective Analysis task
- Create new Objective Analysis task
- Update Objective Analysis task
- Create (Assign) new POC
- Update POC
- Look up POC

Figure 7. Uses cases for a member of the Guidance group.

The third technique is the CRUD technique. To identify the use cases using this technique, we look at each entity of the system data model, and we define the use cases that create the data, read the data, update the data and delete the data. Figure 8 shows an example of the Objective entity use cases.



Data Entity	CRUD	Resulting use case
Objective	Create	Add new objective
	Read	Look up objective produce objective list
	Update	Update Objective information
	Delete	Delete objective

Figure 8. The Objective entity use cases identified by the CRUD technique.

By using both the CRUD and the user goal techniques, we identified all the use cases the *iFRE* system must perform. Table 3 summarizes the CRUD options for all the 15 entities per each group of users.

User Entity	Guidance Group	Data Control Group	Analysis Group	Admin Group
Focus_Area	C,R,U,D	R	R	C,R,U,D
COI	C,R,U,D	R	R	C,R,U,D
Objective	C,R,U,D	R	R	C,R,U,D
Objective Analysis	C,R,U,D	R	R	C,R,U,D
POC	C,R,U,D	R	R	C,R,U,D
Objective_Question	R	C,R,U,D	R	C,R,U,D
Measure	R	C,R,U,D	R	C,R,U,D
System	R	C,R,U,D	R	C,R,U,D
Method	R	C,R,U,D	R	C,R,U,D
Data	R	C,R,U,D	R	C,R,U,D
Event	R	C,R,U,D	R	C,R,U,D
Executed_Event	R	C,R,U,D	R	C,R,U,D
Result	R	C,R,U,D	R	C,R,U,D
Objective_Result	R	R	C,R,U,D	C,R,U,D
Objective_Question_Result	R	R	C,R,U,D	C,R,U,D

Table 3. CRUD options per entity and for each group of user.

In addition to the activities identified by the use cases above, there are also other activities that the system must perform. For the users to achieve their task, they need some tools to allow an efficient coordination. For instance, at any time during the experiment planning, execution or reporting, the users may need to exchange ideas, discuss issues, coordinate meetings or events and share documents. These collaboration tools among others are considered and are added to the *i*FIRE system functionalities. Figure 9 shows an example of the use cases that represent those activities the system will perform throughout the collaboration tools.

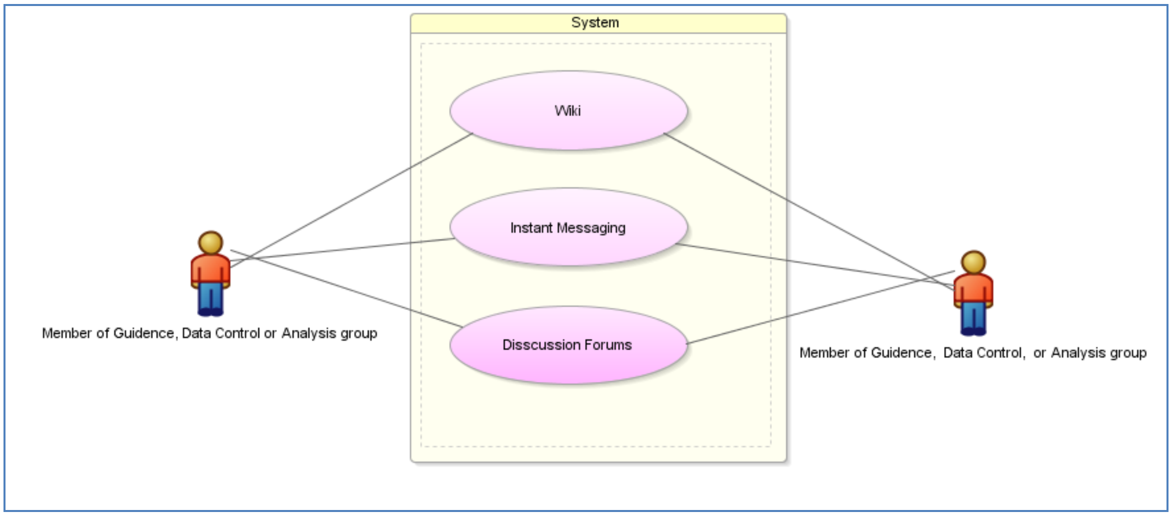

System
Wiki
Instant Messaging
Disscussion Forums
Member of Guidence, Data Control or Analysis group
Member of Guidence, Data Control, or Analysis group

Figure 9. Use cases for the collaboration tools.

3. Experiment Objective Planning Thread Scenario

One of the scenarios *i*FIRE performs is the objective planning thread scenario. This scenario involves different actors and is accomplished through the following steps:

- A member of the Guidance Group creates a new Focus Area, new COI, and an objective (action1).
- A member of the Guidance Group creates an Objective Analysis Task and assigns one member of the Guidance Group as a POC (action2).

- A notification is sent out through the system collaboration tools (e.g.,, announcement, instant messaging) to the Data Planning Group following actions 1 & 2 informing the Group of the creation of an objective, providing them with a POC for that objective, and allowing them to perform actions 3 & 4 for the created objective.
- A member of the Data Planning Group creates an objective question for a given objective (action 3) and one or more measures for that objective question (action 4).
- Members of the Guidance Group collaborate with members of the Thread Planning Group using the system collaboration tools.
- A member of the Data Planning Group modifies an Objective after receiving permission from the POC of that objective.
- A member of the Admin Group adds/modifies/deletes records in all schema tables.

a. Workspace Requirements

To reach the intended outcome, the system should provide different collaboration services to all users and during the different phases of the experiment. These services will provide user coordination, information sharing, and personal productivity tools. The different services are categorized in three groups:

- Social networking services: announcement, discussion, wiki and blog and instant messaging.
- Sharing services: documents, links, and events.
- Personal productivity services: emails, notifications, recent activities, and search capability.

b. Security Requirements

The application deployment is beyond the scope and time available for this research. Therefore, we discuss here security at the application level only. That is, what are the required multiple levels of security for the application. In other words, what level of security is required for each group of users as defined previuously.

Our security requirements analysis led to the following conclusion:

- All users must be garanted access to view of all application content.

- Based on the experiment coordination elements per phase, only those who are given a specfic previlige would be able to add, edit or delete information from the system. This means that security will be defined at the entity level.

D. SUMMARY

In this chapter, we discussed the requirements analysis necessary for the developemnt of *i*FIRE. We divided them into several areas, including the data model, business process, work space, and security requirements. We also described the five phases of the experiment coordination, as well as the actors in each phase. In Chapter IV, we discuss the developement approach and the tools used for implementing the prototype.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DEVELOPMENT APPROACH AND TOOLS

This chapter gives a brief description of the tools and the development approach used to create the web-based application for this thesis. We start by discussing Oracle fusion, which is the umbrella under which all these tools reside. Then, we describe our development approach, Model-View-Controller, and Oracle ADF which is the framework that we chose to develop the prototype. This discussion of Oracle ADF includes an examination of its four layers. A description of the main integrated development environment tool, JDeveloper, follows. We then describe Oracle WebCenter Portal and its main components. Finally, we describe the database modeling and management tools, Oracle SQL Developer Modeler and Oracle SQL Developer.

A. ORACLE FUSION

Oracle Fusion, originally known as Project Fusion, is a set of Oracle business applications and technologies. Oracle, commonly recognized as a database company, has not only expanded its database business considerably, but has also introduced new development tools (such as Oracle Forms and Oracle Reports), and new business applications known as Oracle E-Business Suite. The latter includes business applications such as Human Resources (HR), Financial, and Customer Relation Management (CRM) and others [10].

Oracle's growth helped it to become very successful. This successful growth was furthered by a series of major acquisitions starting with PeopleSoft in 2004 and Siebel Systems in 2005 and followed by Hyperion Solutions, BEA Systems and Sun. These companies brought not only new employees and customers to Oracle but also leading-edge technology solutions such as business intelligence, application server, and middleware products [10].

New acquisitions on one hand brought new technologies, tools and business applications to Oracle, while on the other hand offered new challenges also. Oracle's biggest challenge was how to plan for Oracle's next generation of business applications by making effective use of a variety of applications and technologies. These new challenges were considered by Oracle as new opportunities to develop the next-generation business applications. Oracle capitalized on new technologies (from the acquisitions) and architectures to modernize and redevelop standard business applications. The supporting technology infrastructure and the applications development effort was termed Project Fusion .Later on, only the term “Fusion” was used to represent any development effort or its building blocks [10]. The brand name “Fusion” is normally used with following three technology pillars: Oracle Fusion Applications, Oracle Fusion Middleware and Oracle Fusion Architecture.

1. Oracle Fusion Applications

Oracle’s next-generation business applications such as CRM, Financials, Procurement, and Supply Chain Management are called Oracle Fusion Applications. These applications are built to meet business needs and are available in the form of modules. Similarly, the term Fusion Application could be used to name any application developed by customers using the same principles and technologies.

2. Oracle Fusion Middleware

Oracle Fusion Applications are mainly web-based applications that need application servers and infrastructure in order to run. Oracle Fusion Middleware offers solutions such as Web servers, application servers and content management systems, and offers complete support for development, deployment, and management [11]. It has a wide range of tools and services from a Java Enterprise Edition (Java EE) compliant environment, including development tools, integration services, business intelligence, collaboration and content management [11]. Figure 10 summarizes the Oracle Fusion Middleware solution.

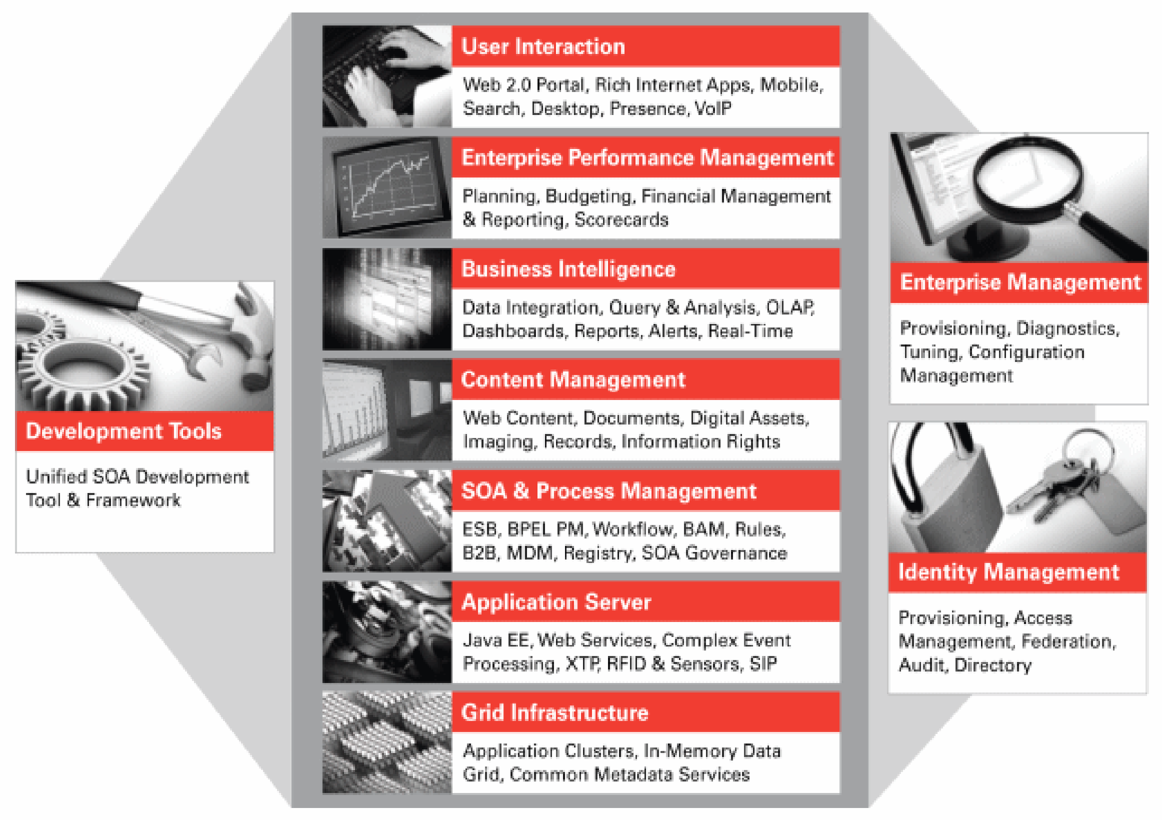

User Interaction
Web 2.0 Portal, Rich Internet Apps, Mobile, Search, Desktop, Presence, VoIP
Enterprise Performance Management
Planning, Budgeting, Financial Management & Reporting, Scorecards
Business Intelligence
Data Integration, Query & Analysis, OLAP, Dashboards, Reports, Alerts, Real-Time
Content Management
Web Content, Documents, Digital Assets, Imaging, Records, Information Rights
SOA & Process Management
ESB, BPEL PM, Workflow, BAM, Rules, B2B, MDM, Registry, SOA Governance
Application Server
Java EE, Web Services, Complex Event Processing, XTP, RFID & Sensors, SIP
Grid Infrastructure
Application Clusters, In-Memory Data Grid, Common Metadata Services
Development Tools
Unified SOA Development Tool & Framework
Enterprise Management
Provisioning, Diagnostics, Tuning, Configuration Management
Identity Management
Provisioning, Access Management, Federation, Audit, Directory

Figure 10. Overview of the Oracle Fusion Middleware solution. From [11].

3. Oracle Fusion Architecture

Oracle Fusion Architecture is the blueprint for building Fusion Applications based on industry best practices and technologies. These blueprints include principles such as service-oriented architecture (SOA), Java Platform, Enterprise Edition (Java EE), model-based pattern and a number of other concepts. Oracle Fusion Applications are built on top of the Oracle Fusion Middleware technology stack.

B. MODEL-VIEW-CONTROLLER

Model-View-Controller (MVC) is a design pattern used in the development of web-based applications. MVC pattern divides the application into three distinct layers: the model, the view, and the controller. These layers play the following roles [12]:

- The model is responsible for accessing and managing the application data. It contains the business logic and functions that manipulate the business data.
- The view renders the User Interface (UI). It manages the presentation aspect of the application based on the model data and raises events (requests) to the controller layer.
- The controller is responsible for the application pages flow and events handling. It accepts and intercepts the user requests and controls the model and view to function accordingly.

Figure 11 depicts the role of these three layers.

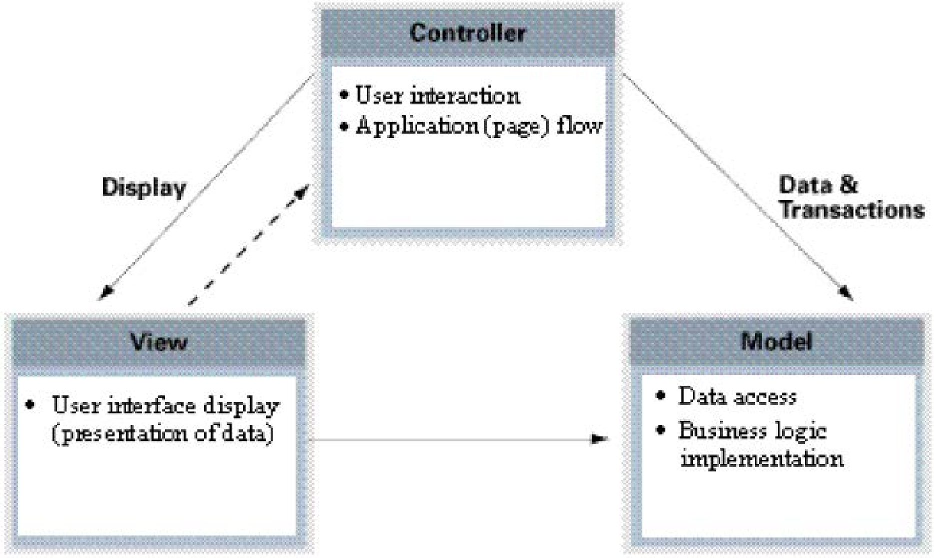

Controller
• User interaction
• Application (page) flow
Display
Data & Transactions
View
• User interface display (presentation of data)
Model
• Data access
• Business logic implementation

Figure 11. The MVC design pattern layers and their roles. From [12].

Use of the MVC pattern helps streamline the application development process, promotes reuse and increases maintainability. The separation of the model and the view allows one group of developer to work on the data model, while another group of developers is working on the UI. It is also possible to design multiple views from the same model.

C. ORACLE ADF

1. Oracle ADF Overview

Many large enterprise applications run on the Java EE platform. Such organizations choose this platform because it accommodates their increased need to build robust, maintainable, and agile applications. Java has improved considerably, but building massive and complex applications with rich UIs still requires significant effort and time from developers. As stated in the previous section, the MVC pattern helps streamline the development process, but developers still require a high level of expertise and knowledge of the technology to build complex applications. They also need tools and a framework to help them organize the application development environments and to keep track of the numerous data sources, components and pages of the applications [13].

Oracle ADF is a complete framework built based on Java Platform, Enterprise Edition (Java EE) to simplify next-generation enterprise application development by providing out-of-the-box infrastructure services and a visual and declarative development experience [14]. It groups the different services and features of different open-source technologies in a single framework [14]. By adhering to standard patterns such MVC and best practices, Oracle ADF abstracts the complexity of the Java EE platform and greatly reduces effort. Oracle ADF provides the applications' infrastructure implementation as part of the framework; thus it minimizes the need for writing code to implement these infrastructures, which allows the developers to focus on the features of the actual application.

Oracle ADF implements the three layers of the MVC design pattern and offers an integrated solution that covers areas such as "Object/Relational mapping, data persistence, reusable controller layer, rich Web user interface framework, data binding to UI, security and customization" [15]. Moreover, Oracle ADF integrates with the Oracle Service Oriented Architecture (SOA) and WebCenter Portal frameworks which simplify the development of complete composite applications. Oracle JDeveloper is the development tool that integrates Oracle ADF elements to provide an environment that covers the full development lifecycle from design to deployment. Oracle JDeveloper’s features are described in Section D.

2. Oracle ADF Architecture

Oracle ADF implements the MVC architecture and further separates the model layer from the business services to enable SOA development. Therefore, Oracle ADF architecture is based on four layers: the business services, model, controller and view. The main building blocks of Oracle ADF for applications development are ADF Business Components, ADF Model, ADF Controller, and ADF Faces. Figure 12 shows the different technologies available for developers for each layer when developing an Oracle ADF application. Through its data binding, the Oracle ADF model layer acts as the glue that integrates the various application components of the controller and the business services layers; thus it makes development flexible. The Data services could be any data sources.

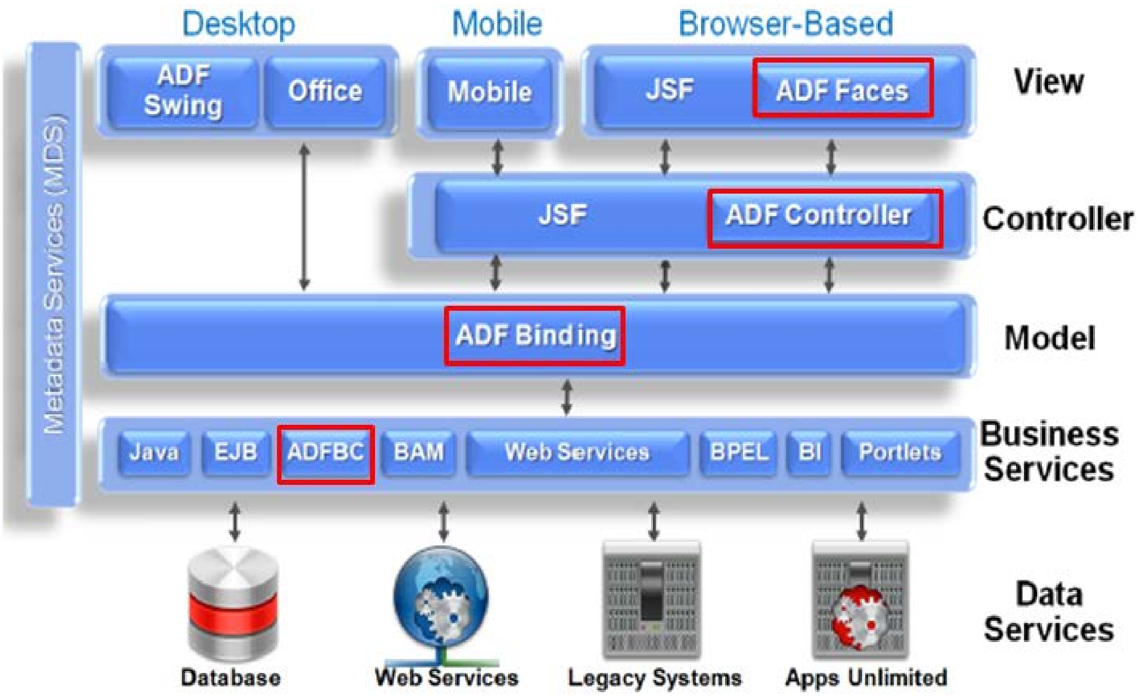


Figure 12. Oracle ADF Architecture layers, Building blocks and the technologies available in each layer. From [15].

The remainder of this section describes the four layers and respectively the four main building blocks of Oracle ADF.

a. *Business Services Layer*

The Business Services layer provides the business logic and handles the data access. The business services implementation is simplified by the ADF Business Components, which free the developer from writing the application’s infrastructural code. ADF Business Components (ADF BC) consist mainly of the Entity Object, View Object and Application model. These components are described below.

(1) Entity Object.

The Entity Object (EO) represents a row in a database table and handles create, update, and delete operations. EO is the ADF BC mechanism for persisting data. An EO maps directly to a database table and provides a data cache for that table. EOs holds the business logic and data validation.

(2) View Object.

The View Object (VO) represents an SQL query from an EO. It defines and filters the specific data (attributes) required by the application. It can be linked by View Links to other view objects to create master-detail hierarchies. VOs handle read operations from the database source and cooperate with EOs for the validation and persistence of data according to change requests made by users.

(3) Application Module.

An Application Module (AM) is a transactional component that plays the role of a work unit–related container. Usually, each container represents a specific business use cases and is the access point to the application data. An AM contains instances of the VOs that are related to those business use cases as well as the required procedures and functions (e.g.,, create, delete) for the user to accomplish them.

Figure 13 shows an example of the ADF BC elements. Three Entity Objects: Customers, Orders and Employees are mapped to three database tables. Two Entity Views are made throughout two SQL queries from the Entity Objects. Finally, the View Objects instances that define the data model and transactions for a particular business task, are grouped in one Application Module.

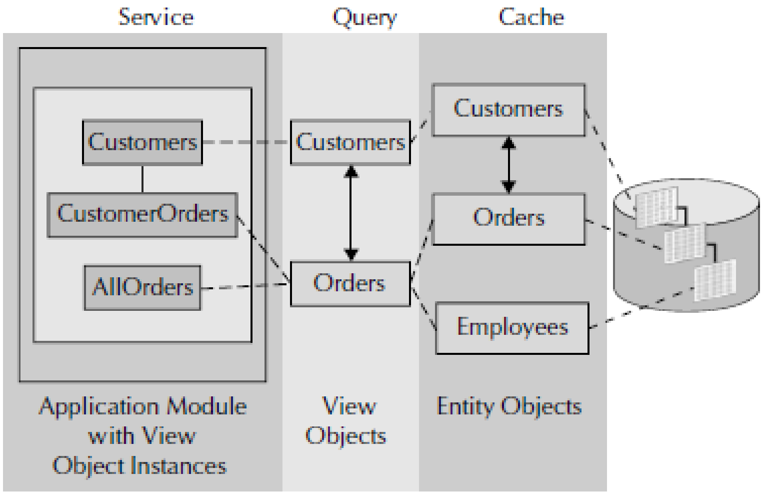


Figure 13. ADF Business Components consists of entity objects, view objects, and application modules. From [14].

b. Model Layer

The model layer binds the business services layer to the view layer, abstracting the implementation details. The ADF model has the following two sub-components:

- **Data Controls:** Abstract the business services access through data bindings and make the implementation of a service transparent to the application developer.
- **Data Bindings:** bind the view components to the appropriate data from the business services.

c. Controller Layer

The controller layer manages the flow of the application pages. In this layer, the ADF controller intercepts and interprets the user input and then commands the model and/or the view to change as appropriate. The ADF controller is extended from the Java Server Faces (JSF) controller to support a modular approach for the application flow control. The key component of the ADF controller layer is the task flow. The ADF task flow breaks the application into a series of tasks instead of representing the application as a single large JSF page flow. Breaking the application flow into a series of task flows offers many benefits, such as simplicity and reusability. Table 4 shows the advantages of the ADF task flow over the JSF page flow.

JSF Page Flow	**ADF Task Flow**
The entire application must be represented in a single page navigation file (`faces-config.xml`). Although you can have multiple copies of `faces-config.xml` in a project, the application loads these files as one at runtime.	The application can be broken up into a series of modular flows that call one another
All nodes within a JSF page flow must be JSF pages. No other types of objects can exist within the JSF page flow.	You can add to the task flow diagram nodes such as views, method calls, and calls to other task flows.
Navigation is only between pages.	Navigation is between pages as well as other activities, including routers.
Application fragments cannot be reused.	ADF task flows are reusable within the same or an entirely different application. After you break up your application into task flows, you may decide to reuse task flows containing common functionality.
There is no shared memory scope between multiple requests except for session scope.	Shared memory scope (for example, page flow scope) enables data to be passed between activities within the task flow. Page flow scope defines a unique storage area for each instance of a bounded task flow.

Table 4. ADF Task Flow Advantages. From [16].

Task flows represent the business process using pages or page fragments through diagrams [13]. There are two types of task flows: bounded and unbounded. An ADF application has only one unbounded task flow that contains the application’s entry point. The bounded task flow represents a piece of reusable business logic. It has a single entry point (default activity view), which is the first access point to the task flow from the browser and zero or more exit points [16]. The unbounded task flow has a set of activities, control flow rules, and managed beans that interact to allow the completion of a specific task. It allows “reuse, parameters, transaction management, reentry, and can render within an ADF region in a JSF page” [16].

Figure 14 depicts an example of a simple bounded task flow diagram that we call AddNewCustomerTF. This task flow represents the process of performing the task of adding a new customer. It has two view activities: Customer and AddCustomerForm and two Control Flow Cases labeled “Add” and “Save” (the two lines terminated by arrows). The view activities define the pages, and the control flows define the navigation between these pages. The Customer view is surrounded by a green circle to indicate it is the default view of the task flow and represents the entry point of the task flow.

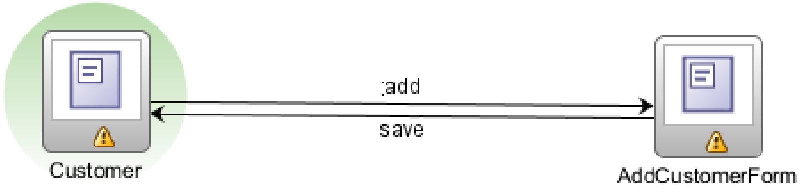


Figure 14. An example of a task flow diagram.

AddNewCustomerTF is created using JDeveloper where the developer simply drags and drops the activity views and the flow control cases from the Components Palette into the task flow template. The developer then double clicks on the activity view icon to create a page fragment where he or she places a form or a table. The navigation between the view activities is based on the action resulting from an event, which could be the user clicking either on a link or a button. Once completely developed, a bounded task flow is saved in the application resources and could be then simply dragged on a specific region of a JSF page. That region contains its own set of navigable pages defined, but the task flow (pages defined under the view activities) allowing users to view different pages and perform many functions without leaving the main page. Bounded tasks can be reused and called by another bounded task flow to perform its specific task.

d. View Layer

The view layer supplies the application’s UI components for displaying and interacting with the application data. In this layer, the technology used to build rich UI is the ADF Faces Rich Client, which is usually abbreviated as ADF Faces. ADF Faces is a UI framework built on the top of the JSF technology with additional features that are not available in the core JSF [10]. ADF Faces has a collection of more than 150 components that facilitate the building of a rich web UI [13]. These components fall into one of the following categories [13]:

- **Layout components:** used for arranging the contents of the page.
- **Input components:** allows users to enter data and accept their input.
- **Structured data components:** includes table and tree components to display tabular or hierarchical data with sorting and filtering functionality.
- **Output components:** used to display text, graphics, icons and for playing audio and video clips.
- **Pop-up components:** includes pop-up browser windows to display pages and page fragments.
- **Menu and toolbars components:** used to launch application actions.
- **Navigation components:** include buttons, links, and menus used to provide access to a specific page or to interact with more complex page flows.

- **Query components:** used to build search forms that support single item search and multiple criteria search.
- **Data visualization components**: include graphs, gauges, pivot tables, Gantt charts and geographic maps.

3. Oracle ADF Benefits

Below is a summary of Oracle ADF key characteristics and features that make it particularly suitable for the development of any Web application [15]:

- **End-to-End Solution** – Oracle ADF provides an integrated and complete solution covering all the Java EE layers from the view layer and data-bindings through the business services and data access. Oracle ADF has mature, complete, and flexible tools to meet specific requirement of various development life cycles phases
- **Development Environment** – Oracle ADF has strong integrated support offered by Oracle JDeveloper. The visual aids and declarative approach provided by JDeveloper reduce the need to write framework code as well as the learning curve for developers. The visual and declarative development experience enhances the developer's productivity and the quality of the software code [10].
- **Platform Independence** –Oracle ADF runtime can be installed on numerous Java EE compliant application servers, and the business services can connect to any database compliant with SQL-92
- **Technology Choice** – Oracle ADF supports multiple technologies for each of the layers (Figure 12). The Oracle ADF Model layer can use the Enterprise JavaBeans (EJB), Web Services, JavaBeans objects as Business Services. Also, the view layers can use UI components implemented using JSF, Desktop Swing applications and MS Office front ends, as well as interfaces for mobile devices.
- **Technology Commitment** - Oracle ADF is the technology choice for building Oracle Fusion Applications, and therefore, it is a committed, supported and consistent technology stack.
- **Metadata-Driven** –Oracle ADF framework layers offer declarative options for development, conFigured from eXtensible Markup Language (XML) metadata, while developers can customize the code wherever needed. Oracle ADF is implemented in Java and thousands of lines of Java code are generated throughout the declarative options (wizards) in JDeveloper. However, the implantation of an application-specific use case is driven by metadata [14]. For example, the navigation between page A and page B is defined in XML, not in Java code, which means that at run time the XML is read to implement this specific navigation [14]. The metadata-driven framework offers many

advantages. First, developers are spared from maintaining the thousands of generated Java codes. Second, it offers clean separation between the metadata implementing the framework features and the Java code written to address a specific business logic, which resides in separate Java classes [14].

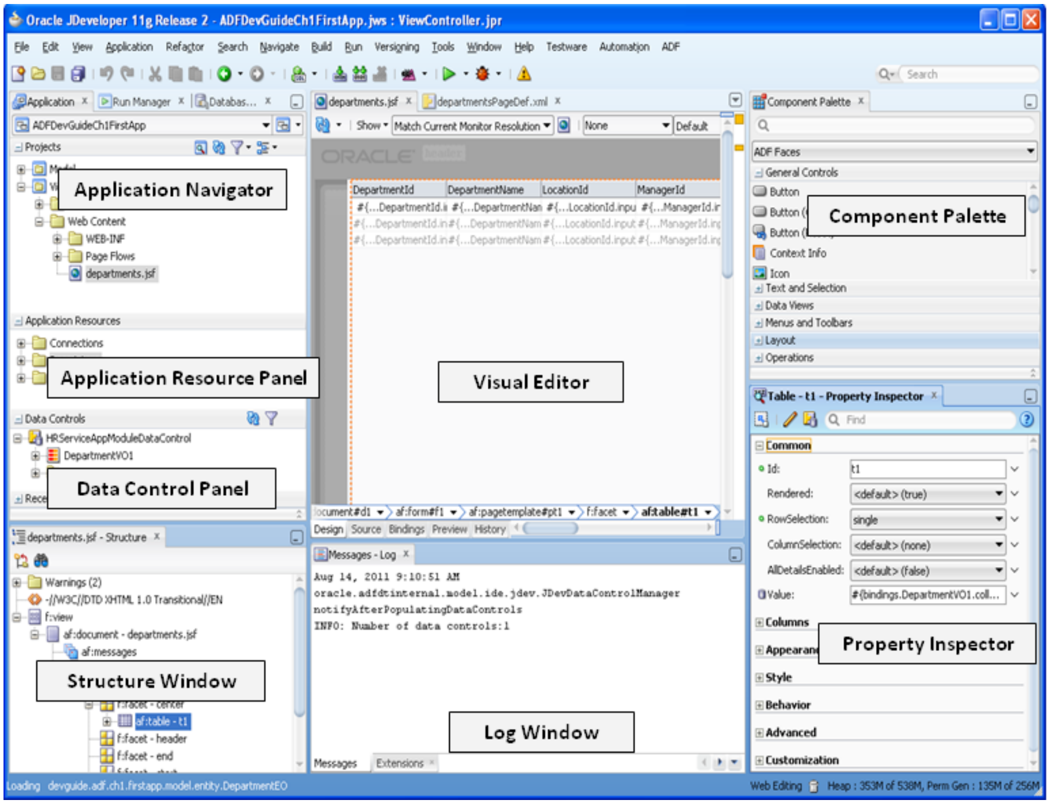
- **Enhanced Reusability** –Oracle ADF and JDeveloper offer high reusability features such as JSF templating, reusable task flows, task flow templating, reusable business services, ADF libraries and JSF fragment based regions.
- **Source availability** - the source code for the ADF framework is available in addition to the declarative and visual editor. This helps developers understand the underlying mechanisms of the framework and debug problems in their applications.
- **Declarative Customization** –Oracle ADF works in conjunction with a MetaData Services (MDS) layer to provide application customization and personalization at run time. The changes at run time are then persisted via the MDS repository.

D. ORACLE JDEVELOPER

Oracle JDeveloper is a free integrated development environment (IDE) and is the main development platform for the Oracle Fusion Middleware. JDeveloper supports the entire development lifecycle with integrated features for modeling, coding, debugging, testing, profiling, tuning and deployment. It has complete tools to support application development from the design to the production. JDeveloper has an integrated WebLogic server (described in the following section) to run and test the application. JDeveloper supports the creation of the most commonly used Unified Modeling Language **(**UML**)** diagrams including use cases, sequences, and activity and class diagrams. JDeveloper provides a full database development environment from design and modeling to implementation, for both Oracle and non-Oracle databases [17]. JDeveloper integrates SQL Developer capabilities to provide database browsing, query execution and object definition and manipulation. JDeveloper supports various technology stacks including Java, SOA, Oracle WebCenter Portal (described in the next section), SQL and PL/SQL, HTML, and JavaScript [16].

JDeveloper is written completely in Java, which means it is platform independent. Thus, it is compatible with Windows, Linux, Mac OS X and other UNIX-based systems. JDeveloper provides a visual and declarative approach; therefore, it simplifies Java EE applications development by providing wizards, editors, visual design tools and deployment tools to create high-quality standard components, including applets, JavaBeans, JavaServer Pages (JSP), servlets and Enterprise JavaBeans (EJB) [10]. JDeveloper also provides a public Add-in Application Programming Interface (API) to extend and customize the development environment to allow integration with external products. JDeveloper gives the developer the option to work directly with the source code at any time. Additionally, JDeveloper supports the creation of templates.

Figure 15 is a snapshot of the JDeveloper interface showing the most commonly used editor windows and tools. These windows and tools are described as follows [10]:


Oracle JDeveloper 11g Release 2 - ADFDevGuideCh1FirstApp.jws : ViewController.jpr
File Edit View Application Refactor Search Navigate Build Run Versioning Tools Window Help Testware Automation ADF
Application Navigator
Application Resource Panel
Data Control Panel
Structure Window
Visual Editor
Log Window
Component Palette
Property Inspector

```
Aug 14, 2011 9:10:51 AM
oracle.adfdtinternal.model.ide.jdev.JDevDataControlManager
notifyAfterPopulatingDataControls
INFO: Number of data controls:1
```

Figure 15. Snapshot of JDeveloper main window. Form[10].

- **Application navigator**: Through this window options, the developer creates the application projects (e.g.,, Model, Portal) and source files, and manages the contents and the resources of applications.
- **Application resource panel**: This window displays the application-level resources such as database connection information, web service presence, instant Messaging and content repository. It also displays the configuration files and metadata files used to configure ADF Business Components.
- **Visual editor**: This window is the area in which the UI is developed. It helps the developer visually build the ADF application UI. It provides a WYSIWYG (What You See Is What You Get) editor. The developer can also switch to the source window to view the source code or to the binding window to review or edit the data binding.
- **Data control panel**: This panel lists commonly used data controls in any application development. The data control panel displays the data collections, attributes, built-in operations (e.g.,, create, insert, save) and business methods that can be dragged-and-dropped on the visual editor window to build the UI. Any time an item is dragged and dropped in the visual editor window, a metadata XML file is generated to bind the business data with the UI.
- **Structure window**: The structure window displays a structural view of the currently selected data in the visual editor window. It is also possible to drag-and-drop components from the data control panel to this window rather than to the visual editor window.
- **Component palette**: The component palette window displays all available components (e.g.,, buttons, panel box, search toolbar, etc.) for page design or navigation definition based on the technology used for application development.
- **Property inspector**: The property inspector displays the exposed properties of the component selected in the structure window or in the visual editor and allows the setting of its appearance or behavior.
- **Log window**: The log window displays the logs from various components such as the compiler, audit rules, debugger, and profiler.

E. ORACLE WEBLOGIC SERVER

Oracle WebLogic server is web application server and is fully compliant with the Java EE standards. The main purpose of using Oracle WebLogic server is to deploy web application. Oracle WebLogic server is one of Oracle Fusion Middleware components and Oracle Fusion applications rely on WebLogic server to run their code. There are many key terminologies that are related to WebLogic Server such as Server Instance, Server Domain, Administration Server, Managed Severs and cluster (Figure 16). These terms are described as follow [18].

- **WebLogic Server Instance**: a Java Virtual Machine (JVM) that runs the Java code. The instance represents the server active component and manages the application resources necessary to its functioning. The server instance receives and processes the client requests, managed the resources related to the request and sends the processed request back to the client.
- **WebLogic Server Domain**: a logically related group of WebLogic Server resources. A domain is a set of WebLogic Server instances including a special WebLogic Server instance called the **Administration Server** which is the primary means of managing the domain. In addition to the Administration Server, the domain contains one or more Managed Servers.
- **Administration Server:** a special WebLogic Server instance and is the central point from which all resources in the domain are configured and managed. It is designed to manage the domain and the managed server rather than running the application.
- **Managed Servers:** are WebLogic Server instances that host business applications, application components, Web services, and their associated resources.
- **WebLogic Server Cluster:** a group of managed servers that run simultaneously to balance loads and increase reliability and availability for critical applications.

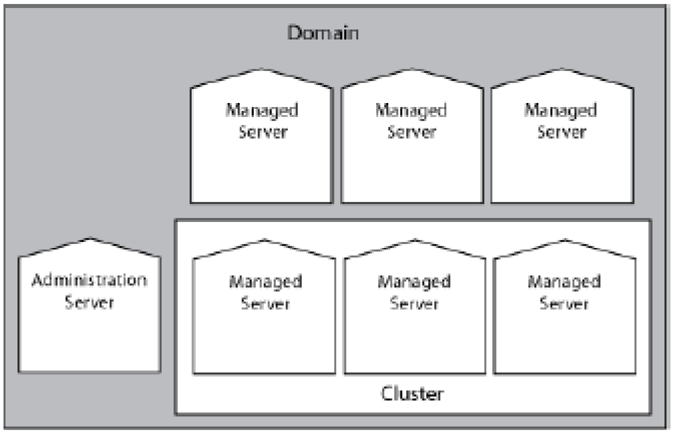


Figure 16. WebLogic domain. From [19].

F. ORACLE WEBCENTER

1. Introduction

Web 2.0, the term used to describe the second generation of the World Wide Web, refers to the transition from static HTML Web pages to more dynamic Websites and applications that allow users to create, share information, collaborate and communicate. Social networking sites, blogs, wikis, video sharing and instant messaging are all examples of Web 2.0. Companies want to leverage the Web 2.0 technologies by integrating them into their intranet, extranet and business processes. The term Enterprise 2.0 was coined to describe the use of Web 2.0 technologies such as the social software and collaborative tools by companies. Enterprise 2.0 is defined as “the use of emergent social software platforms within companies, or between companies and their partners or customers” [20].

Oracle WebCenter is a complete development platform used to build rich, customizable Enterprise 2.0 applications. Throughout its comprehensive suite of components and services, it provides all necessary building blocks to create next-generation Enterprise 2.0 applications and portals [13]. Oracle WebCenter Suite is “the modern user experience platform for the enterprise and the Web” [21]. The suite consists of four main components including: Portals and Websites; Composite Applications; Social and Collaboration, and Content Management (Figure 17). Oracle WebCenter Portal is the main component used in the development of *i*FIRE application. Oracle WebCenter Content Core Capabilities (formerly Oracle Universal Content Management) is implemented for the document management. The remainder of this section (D) will focus on Oracle WebCenter Portal.

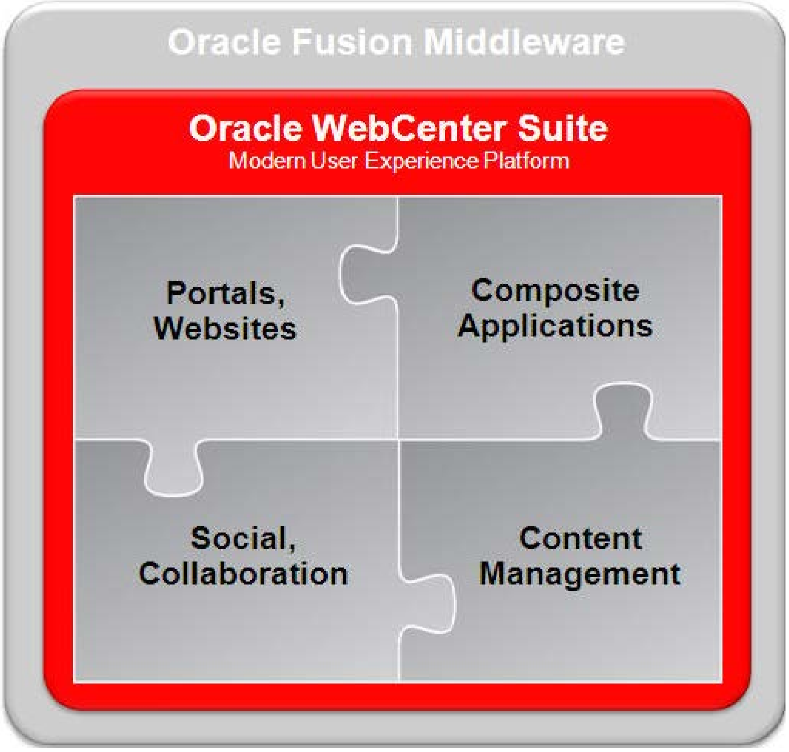


Figure 17. Oracle WebCenter Suite Components. From [21].

2. Oracle WebCenter Portal

Oracle WebCenter Portal is one of the Oracle WebCenter Suite products and is “a comprehensive set of portal-specific components that help organizations build enterprise-scale transactional and composite applications” [21]. Oracle WebCenter Portal consists of many components (Figure 18) that provide a full range of functionality to develop Enterprise 2.0 applications and portals, and to maintain efficient enterprise portals and intranets, and composite applications [22].

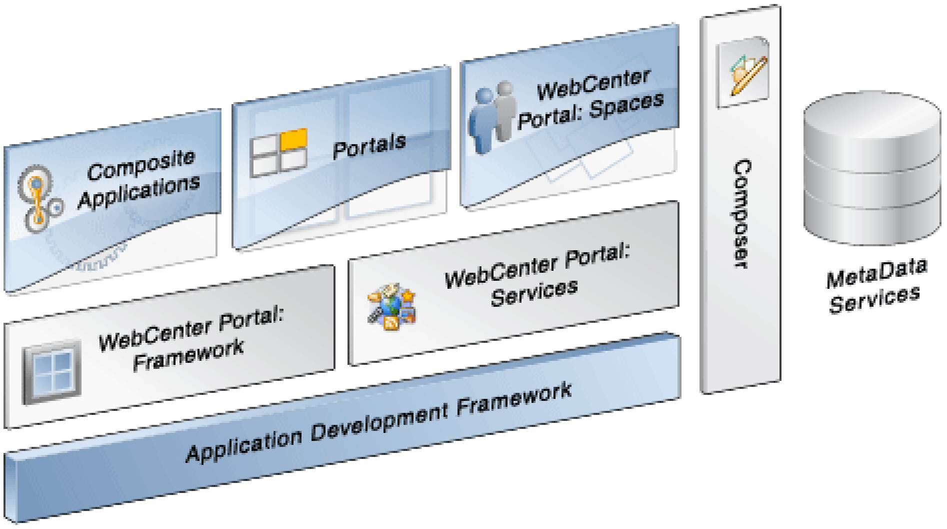
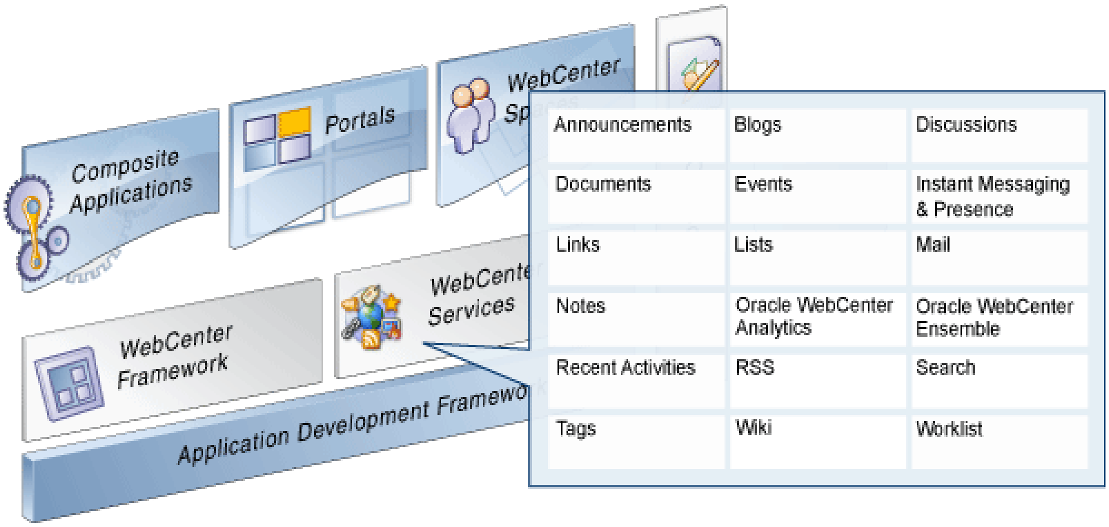

Composite Applications
Portals
WebCenter Portal: Spaces
Composer
MetaData Services
WebCenter Portal: Framework
WebCenter Portal: Services
Application Development Framework

Figure 18. Oracle WebCenter Portal Components. From [22].

These components are described as follows [22]:

- **Oracle WebCenter Portal: Framework:** The framework is delivered as an extension of Oracle JDeveloper and expands the capabilities of the Oracle ADF with enterprise portal capabilities, including run-time personalization and customization. It is a declarative JSF-based framework that enables embedding of AJAX-based components, portlets, services, and content into context-rich customizable enterprise portals. It provides content integration through Content Repository API for Java (JCR) (JSR170) including the Oracle Content Server, file system,.etc. The framework adds many features, such as page hierarchies, navigation models, delegated administration, customization, themes and skins and portal preferences.
- **Oracle WebCenter Portal: Spaces:** A prebuilt (ready-to-use) application that pulls together WebCenter Services in terms of social networking, communication, collaboration and personal productivity. WebCenter Spaces is built using JSF, Oracle ADF, WebCenter Framework, WebCenter Web 2.0 services and Composer. It has all the required tools to create Websites, portals, community, and social networking sites rapidly for thousands of users.
- **WebCenter Portal: Services:** This feature streamlines the integration of new social networking tools with enterprise information and business processes. The list of services includes the following: wikis, blogs, RSS, lists, discussions, announcement, instant messaging, links commenting, sharing, polls, search and more (Figure 19).

- **Composer in WebCenter Portal:** The composer enables runtime customization and personalization of pages that are created with WebCenter Portal Framework or WebCenter Portal Spaces, and includes a set of services to create and manage portal pages at runtime. Runtime customization changes are stored and managed by Oracle Metadata Services (MDS). WebCenter Composer has a resource catalog where task flows can be dropped in, and WebCenter Framework allows the task flows to be dragged and dropped into customizable pages at run time, thus allowing easy add-in business process functionality and easy maintenance.



Announcements	Blogs	Discussions
Documents	Events	Instant Messaging & Presence
Links	Lists	Mail
Notes	Oracle WebCenter Analytics	Oracle WebCenter Ensemble
Recent Activities	RSS	Search
Tags	Wiki	Worklist

Figure 19. Oracle WebCenter Portal: Services. From [22].

G. ORACLE SQL DEVELOPER DATA MODELER

The Oracle SQL Developer Data Modeler is a graphical data modeling tool that simplifies the data modeling development process. Oracle SQL Developer Data Modeler assists and augments communication between data architects, database administrators, application developers and users. SQL Developer Data Modeler can be used in various database data models and has vast application development usage. Oracle SQL Developer Data Modeler is a standalone, independent product that can run on Windows, Linux and Mac OS X. The Oracle SQL Developer Data Modeler is "a complete model-to-implementation solution for data-related modeling, such as adding and implementing new data requirements, and capturing existing database implementation to provide a graphical representation and related metadata documentation" [23].

SQL Developer Data Modeler model has three synchronized layers: a logical model, relational model, and physical model [23]. The logical model includes standard logical modeling tools to draw the entities and relationships; it provides box-in-box presentation for the super-type and sub-type hierarchy. The relational model is created by forward engineering (transformation) of the logical model and all many-to-many relationships and all super-type and sub-type hierarchies are then created [23]. It is the intermediate model between the logical and the physical model. The logical model can be transformed by more than one relational model. From each relational model, the developer can create an unlimited number of physical models that are compatible with the database management system (DBMS) such as Oracle 10g, Oracle 11g, Microsoft SQL Server, and others. The SQL Developer Data Modeler facilitates the generation of the *Data Definition Language* (DDL) script, which in turn will be run to create the database in the database server.

H. ORACLE SQL DEVELOPER

Oracle SQL Developer is a free graphical tool that streamlines the database development tasks. It improves productivity and simplifies management of the Oracle database. Oracle SQL Developer supports browsing of database objects, running SQL statements and SQL scripts, editing and debugging PL/SQL statements [24]. SQL Developer offers complete end-to-end development of PL/SQL applications.

Salient features of SQL developer relative to the application we developed are described below [24]:

1. Managing Database Connections

Users can create a database connection to a database server using a simple wizard dialog. After the establishing the connection, users can browse the database, create schema objects, execute and tune ad-hoc SQL statements, run reports against the data dictionary, as well as create, execute and debug PL/SQL by creating a database connection.

2. Working With the SQL Worksheet

An SQL Worksheet window is automatically opened as soon as a connection is established. Users can then input and run any SQL, PL/SQL, and SQL *Plus commands in that SQL Worksheet. Furthermore, users can specify any actions that can be processed by the database connection associated with the worksheet, such as creating a table, inserting data, creating and editing a trigger, selecting data from a table and saving that data to a file [24].

3. Browsing the Database

Database browsing gives users the ability to rapidly review the definitions of objects in a very-easy way. Using the connections navigator, users can browse through a database schema including tables, views, indexes, packages, procedures, functions, queues and queue tables, triggers, types, sequences, materialized views and materialized view logs, synonyms and public synonyms, database links and directories [24].

4. Producing SQL Scripts

As discussed earlier, the connections navigator allows users to create, edit and update database objects. In addition, users can export the DDL for one or more objects in the schema or for all the objects in the schema. This export DDL option gives the user the ability to create a complete execution script [24].

I. ORACLE DATABASE 11*G* EXPRESS EDITION

Oracle Database 11*g* Express Edition (Oracle Database XE) is a free version of Oracle relational database server. Oracle Database XE is used by developers and database administrators for training and deployment purposes. Oracle Database XE can be installed on any size host machine with any number of CPUs (one database per machine), and can store up to 11GB of user data [25].

J. SUMMARY

In this chapter, we discussed the development approach and described Oracle tools used for the development of *i*FIRE. In Chapter V, we will briefly describe the development process and what role played by each of these tools in the development effort. We will also describe the application functionalities and features.

V. APPLICATION DEVELOPMENT

A. INTRODUCTION

In the previous chapters, we discussed the purpose of *i*FIRE, performed requirements analysis and described the tools used for the design and development. This chapter describes the *i*FIRE application implementation – how the application UI looks like? What it can and cannot do? We first describe the implementation processes. Then, we discuss the application features and functionalities as well as some considerations related to the UI and application functionalities.

B. DEVELOPMENT PROCESS

As described in Chapter IV, numerous Oracle tools were used to develop the *i*FIRE application. The tasks performed using these tools are described in the following two sections. The first section addresses the database implementation, and the second discusses the process implementation (Figure 20).

1. Database Design and Implementation

The tools used for the database design and implementation are Oracle SQL Developer/Data Modeler, Oracle SQL Developer and Oracle Database Express Edition. The development process of the database is described as follow:

- **Oracle SQL Developer/Data Modeler:** is used to develop the data model and generate the ERD and the DDL script of the application database. First, the entities with their attributes and primary identifiers were defined. Then, the relationships between those entities were established, which constitute the database logical model. Second, the relational model is generated form the logical model where the foreign key and constraints are defined. Third, the physical model is generated for the Oracle 11 g database platform. Finally, the DDL script for the application database is exported and saved as a file (Appendix A contains the SQL script).

- **Oracle SQL Developer:** is used to create, populate, and manage the application database. First, we created user/schema in Oracle Database Express (11 g) server. Second, we connected to this user account (***I***FIRE) and ran the DDL script to create the database schema. Third, we imported the application data stored in a spreadsheet file and populated the database tables. Finally, we created sequences and triggers to auto-generate the primary keys of some entities.
- **Oracle Database Express Edition 11*g*** (Release 2): is the database server where the application database resides.

2. Process Implementation

Once the database was created and populated, we started the process implementation using the following tools: Oracle JDeveloper, Oracle Content Server and Oracle WebCenter: Spaces. The implementation process and associated tools are described in the following paragraphs:

- **Oracle JDeveloper** (Release 1): is the IDE used to develop *i*FIRE application. It integrates Oracle ADF and WebCenter Framework. Using JDeveloper, we first created a portal application to generate all the default folders. Second, we connected to the database server and generated the ADF Business Components (Model Layer) from *i*FIRE database. Third, we built the application page template and added the elements of the main navigation bar. Fourth, we created the task flows (using the data control, data binding, Java Beans, ADF faces, etc.) to implement the business logics (use cases) (Appendix B contains screenshots of many of those task flows). Fifth, the users and the application resources security were then defined. Finally, we deployed and tested the application using JDeveloper integrated WebLogic server.
- **Oracle Content Server (UCM server):** was used for content management to allow users to share documents.
- **Oracle WebCenter: Spaces**: was used to create a common space for the user to communicate and collaborate. It offers many services such as lists, messages, calendar, discussion forums, announcement, etc.

Figure 20 summarizes the tools used for the application development and the tasks performed by each.

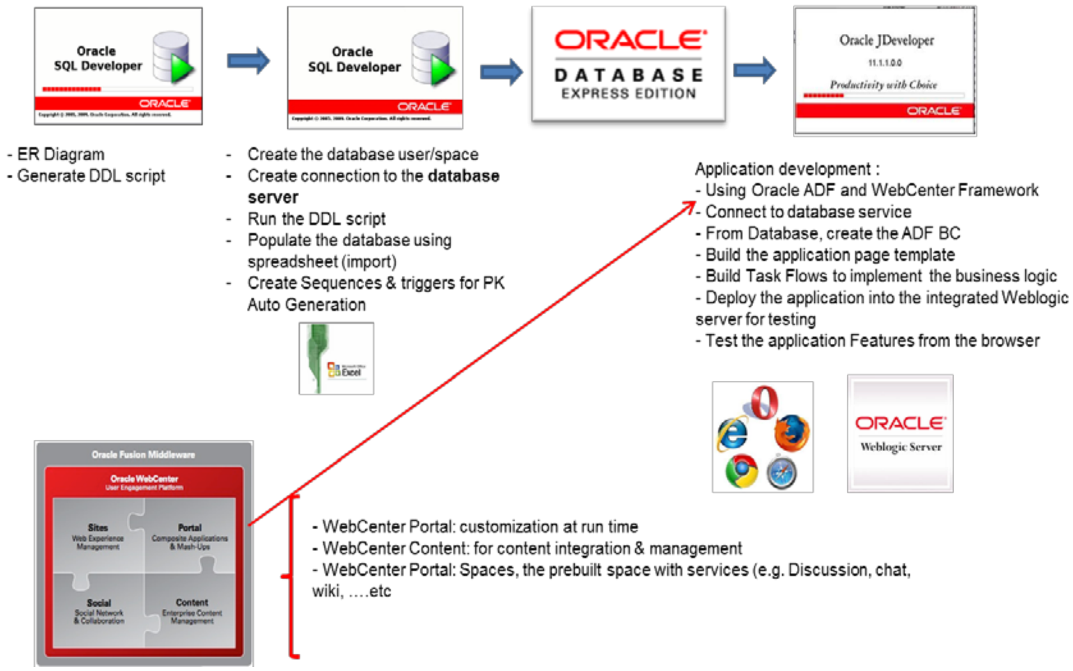

Oracle SQL Developer
ORACLE
Oracle SQL Developer
ORACLE
ORACLE DATABASE EXPRESS EDITION
Oracle JDeveloper
11.1.1.0.0
Productivity with Choice
ORACLE
- ER Diagram
- Generate DDL script
- Create the database user/space
- Create connection to the database server
- Run the DDL script
- Populate the database using spreadsheet (import)
- Create Sequences & triggers for PK Auto Generation
Excel
Application development :
- Using Oracle ADF and WebCenter Framework
- Connect to database service
- From Database, create the ADF BC
- Build the application page template
- Build Task Flows to implement the business logic
- Deploy the application into the integrated Weblogic server for testing
- Test the application Features from the browser
ORACLE Weblogic Server
Oracle Fusion Middleware
Oracle WebCenter
User Engagement Platform
Sites
Web Experience Management
Portal
Composite Applications & Mash-Ups
Social
Social Network & Collaboration
Content
Enterprise Content Management
- WebCenter Portal: customization at run time
- WebCenter Content: for content integration & management
- WebCenter Portal: Spaces, the prebuilt space with services (e.g. Discussion, chat, wiki, ….etc

Figure 20. Development Tools

C. APPLICATION DESCRIPTION

1. Overview

In the design and developemnt of the *i*FIRE application, several design factors were taken into consideration. First, the UI should be simple, easy to use, and self-explanatory. That is, users should easily navigate the application and find what they are looking for. To accomplish this goal, the UI was designed to mirror as closely as possible the five phases of conducting an experiment: Objective planning, Data Planning, Event Management, Result Reporting, and Result Analysis. This approach would make navigating and finding information very intuitive. Second, users should have powerful search capabilties to find the information they are looking for precisely without having to scroll over many pages of information. We therefore implemented search feature where users can find information by using the thread number described in Chapter III section B.1.a. Third, users should not have to remember codes or IDs of the different entities. In fact, most of the entity identifies (e.g., system ID, event ID, Objective NO) are meaningless to the users. Therefore, we implemented a dropdown menu option in most of the application functions. The dropdown option allows the users to use familiar attributes of the different entities rather than cryptic idenfiers/codes to identify entities of interest.

Figure 21 is a snapshot of the application home page. The navigation bar has seven buttons that takes the user to different application pages: the home page, the five pages representing the experiment coordination phases, and the Word Space page. The home page contains the following: the login form, a welcome area where descriptive text could be added, and an image that depicts the experiment five phases and their elements to serve as a road map for the users.

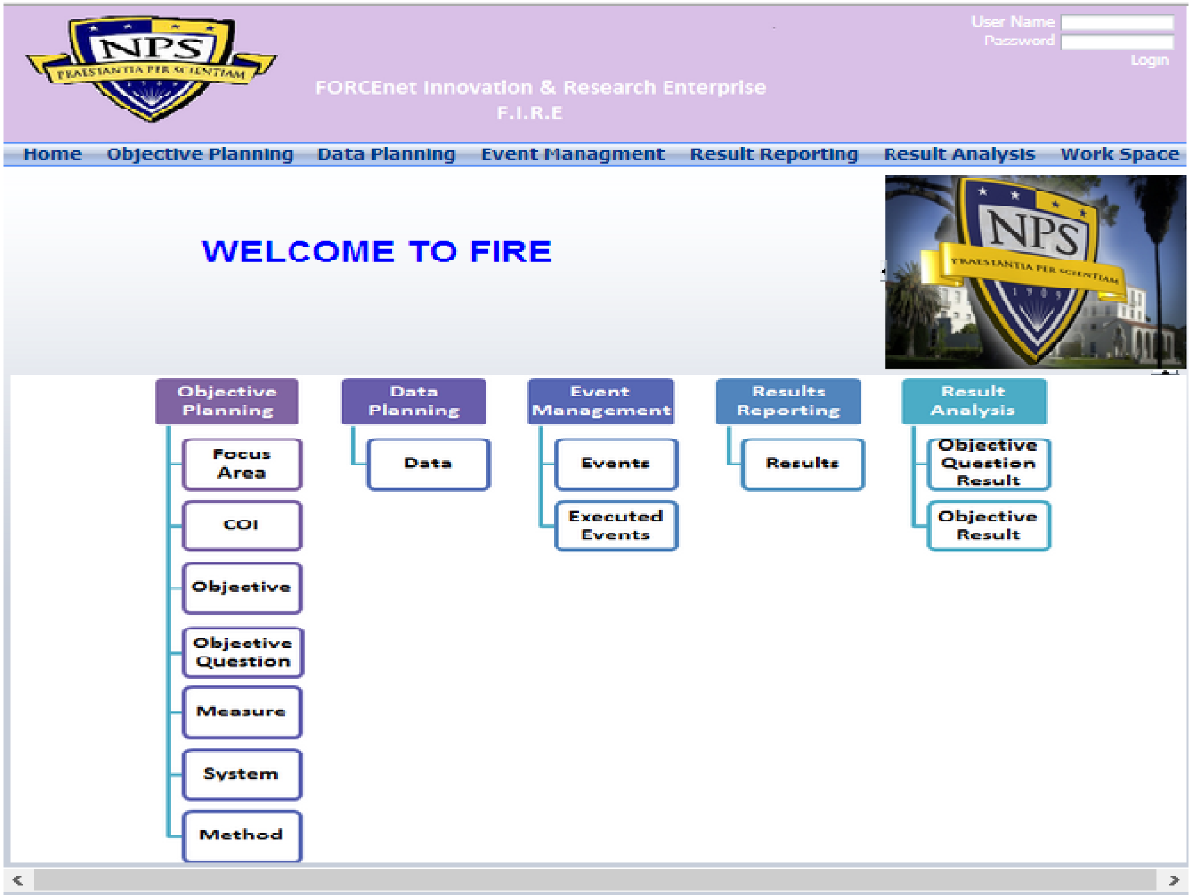
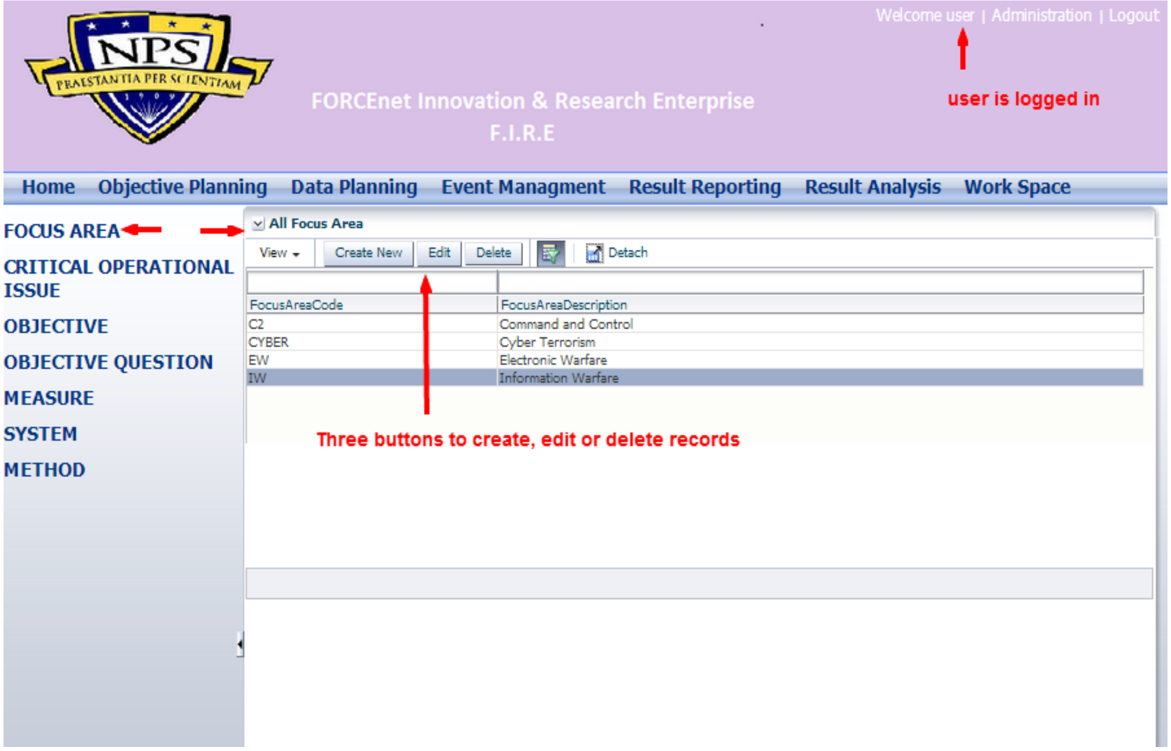


Figure 21. The application Home page

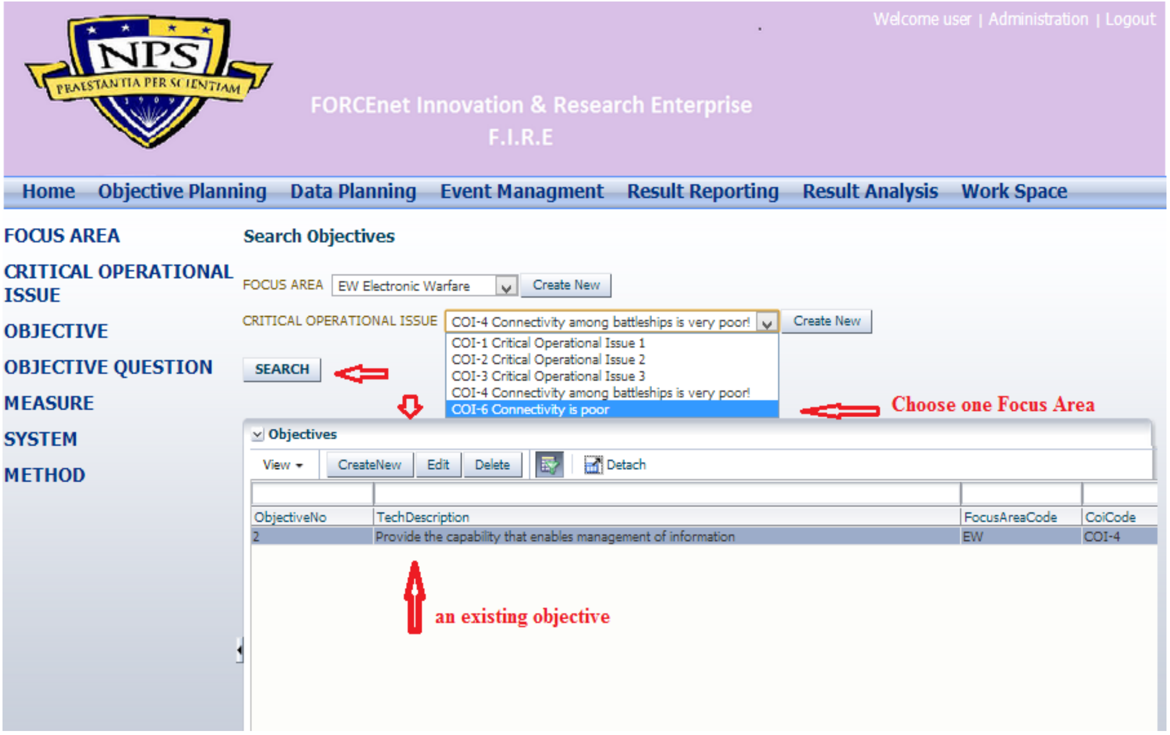
The objective planning page has links (on the navigation bar on the left of the page) to the elements of the objective planning phase (Figure 22). When a user clicks on a link, the appropriate view appears on the right side of the page. For example, if the user clicks on the FOCUS AREA link, a table that displays all the focus area appears on the right side, along with three buttons to create, edit, or delete records. As soon as the user clicks on one of these buttons, a popup page appears that allows him/her to input data, confirm the changes, or cancel the operation. Similarly, for the Critical Operation Issue, System, and Method links, corresponding tables are displayed as well as create, edit, and delete buttons.



FocusAreaCode	FocusAreaDescription
C2	Command and Control
CYBER	Cyber Terrorism
EW	Electronic Warfare
IW	Information Warfare

Figure 22. Snapshot of the FOCUS AREA link output

For the Objective, Objective-Questions, and Measures, we added a multi-criteria search feature. Figure 23 depicts the display resulting from clicking on the Objective link. Using a drop down menu, a user can select a specific Focus Area and Critical Operational Issue, and then decide whether to create a new objective or edit or delete an existing one (Figure 23). As stated earlier, the dropdown menu contains the code/number and description of the corresponding Focus Area or Critical Operation Issue to simplify the user tasks, and exempts the user from remembering the identifying codes/numbers of these entities. A smilar search feature, and resulting display and control buttons, are available in the other four pages: Data Planning, Event Management, Result Reporting, and Result Analysis. The following section provides a full scenario (Use Case) to demonstrate the functionality of the application.



FOCUS AREA: EW Electronic Warfare

CRITICAL OPERATIONAL ISSUE: COI-4 Connectivity among battleships is very poor!

- COI-1 Critical Operational Issue 1
- COI-2 Critical Operational Issue 2
- COI-3 Critical Operational Issue 3
- COI-4 Connectivity among battleships is very poor!
- COI-6 Connectivity is poor

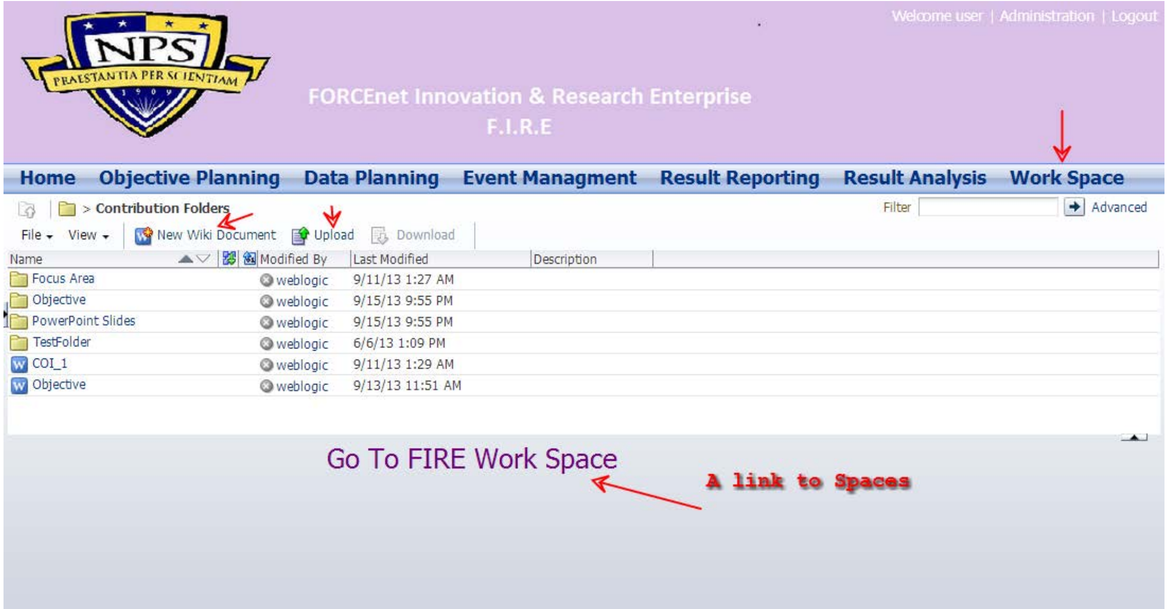
Choose one Focus Area

ObjectiveNo	TechDescription	FocusAreaCode	CoiCode
2	Provide the capability that enables management of information	EW	COI-4

an existing objective

Figure 23. A snapshot of the Objective link view.

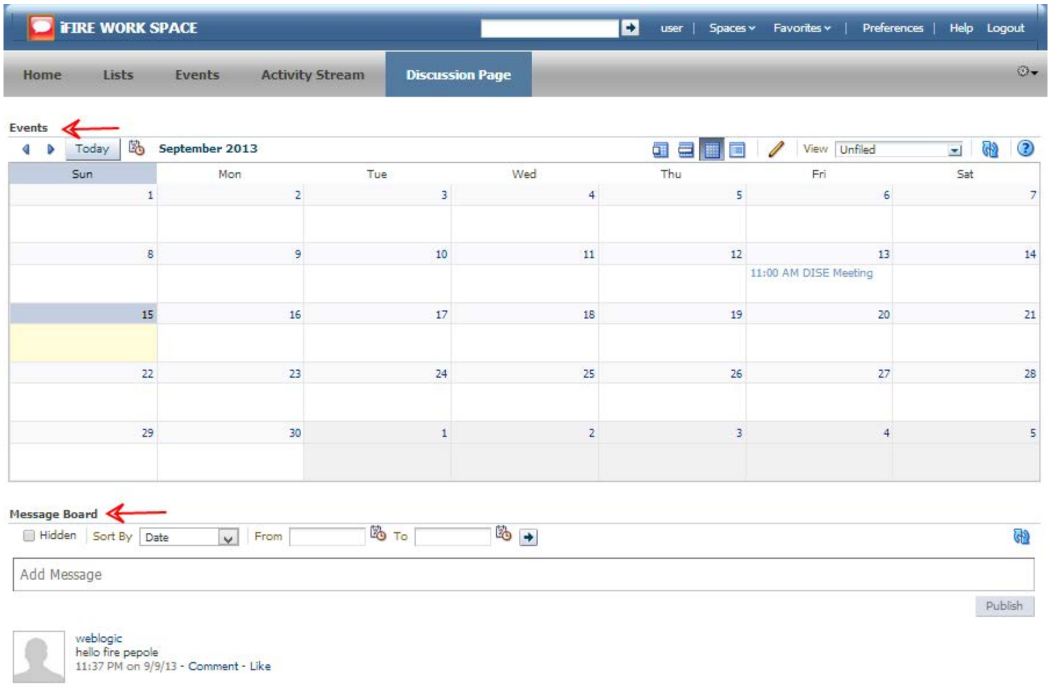
The last button in the navigation bar, Work Space, directs users to the document management feature of the application where they can easily share information and documents. As shown in Figure 24, the Work Space section enables users to create new folders, upload files and create wikis.



Name	Modified By	Last Modified	Description
Focus Area	weblogic	9/11/13 1:27 AM	
Objective	weblogic	9/15/13 9:55 PM	
PowerPoint Slides	weblogic	9/15/13 9:55 PM	
TestFolder	weblogic	6/6/13 1:09 PM	
COI_1	weblogic	9/11/13 1:29 AM	
Objective	weblogic	9/13/13 11:51 AM	

Figure 24. A snapshot of Work Space page.

At the bottom of the Work Space page, a link takes the user to the *i*FIRE Space (Figure 25), implemented using the prebuilt spaces component available from WebCenter Portal: Spaces. We added/configured some services such as messages, lists, activity streams, and events.

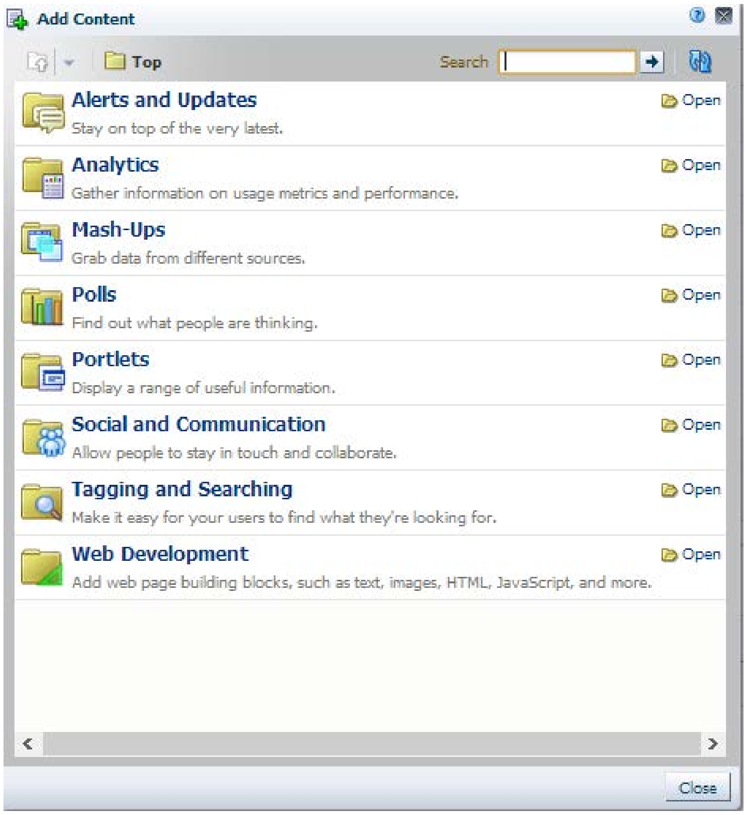


Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13 11:00 AM DISE Meeting	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

weblogic
hello fire pepole
11:37 PM on 9/9/13 - Comment - Like

Figure 25. A snapshot of one the *i*FIRE Space pages.

There are many others features/services that can be added to the space, such as blogs, discussion forums, announcements, and Instant Messaging. Many of these services require the installation and configuration of special servers, which has not been done due to the limitation in time, but can easily be done in the production server. Figure 26 shows the categories of the services/features available for the space and that can be added at run-time.



Alerts and Updates
Stay on top of the very latest.

Analytics
Gather information on usage metrics and performance.

Mash-Ups
Grab data from different sources.

Polls
Find out what people are thinking.

Portlets
Display a range of useful information.

Social and Communication
Allow people to stay in touch and collaborate.

Tagging and Searching
Make it easy for your users to find what they're looking for.

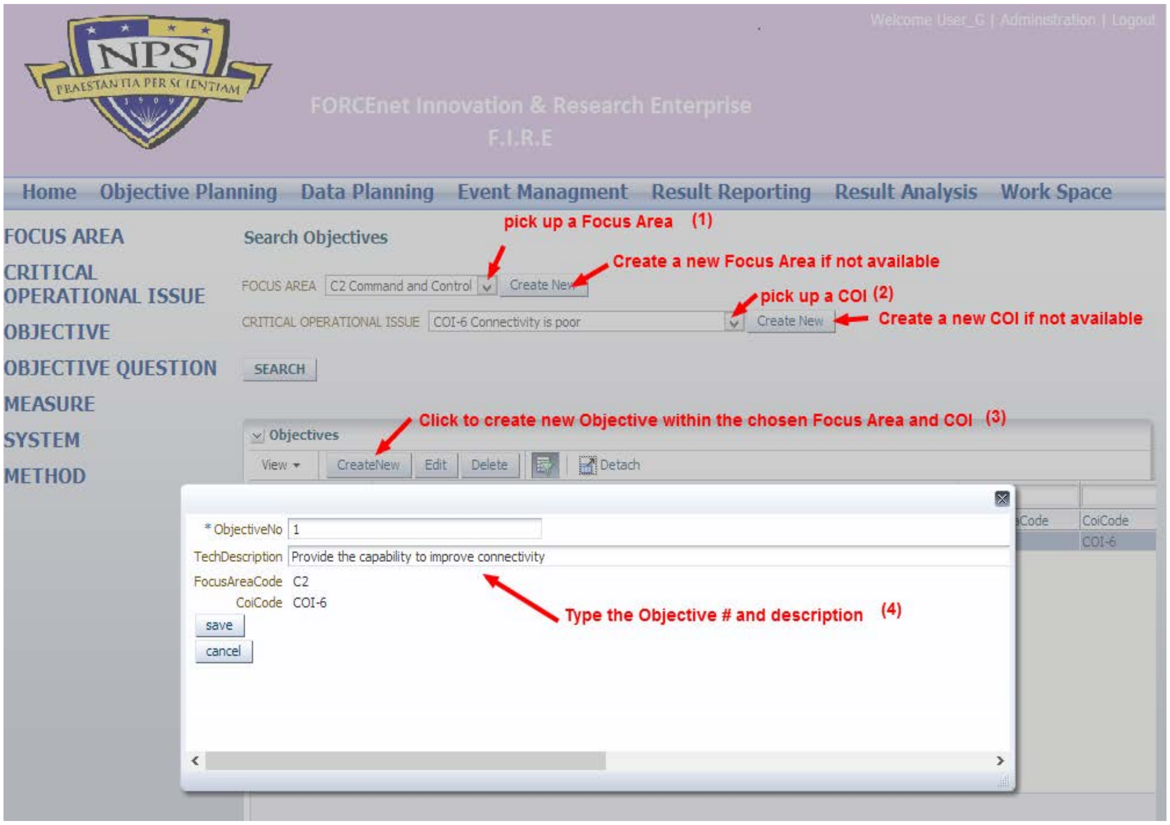
Web Development
Add web page building blocks, such as text, images, HTML, JavaScript, and more.

Figure 26. A snapshot of the services/features that can be added to *i*FIRE Space at run-time.

2. Functionalities/Scenario

The best way to understand the application functionalities is through a scenario. In this section, we take the reader through a comprehensive scenario (Use Case) that covers the five phases of the experiment coordination. Functions in this scenario are performed by users from the three groups (defined in Chapter III): Guidance group, Data Control group and Analysis group. The scenario consists of the following steps:

- **Step 1:** A user of the Guidance group creates an objective within a specific focus area and COI. After clicking on the Objective link under the Objective Planning page, the user simply selects the desired focus area (C2) and COI (COI-6) from the dropdown menu, then, clicks on the CreateNew button (Figure 27). A popup window appears containing a blank form with the objective attributes fields. The user proceeds with filling in the Objective attributes (e.g., ObjectiveNO: 1) and then clicks on the Save button. The user has the option to cancel the operation by simply clicking on the Cancel button. The user also has the option to create a new Focus Area and/or a new COI, if they do not already exist.



pick up a Focus Area (1)

Create a new Focus Area if not available

pick up a COI (2)

Create a new COI if not available

Click to create new Objective within the chosen Focus Area and COI (3)

Type the Objective # and description (4)

FOCUS AREA: C2 Command and Control

CRITICAL OPERATIONAL ISSUE: COI-6 Connectivity is poor

* ObjectiveNo: 1

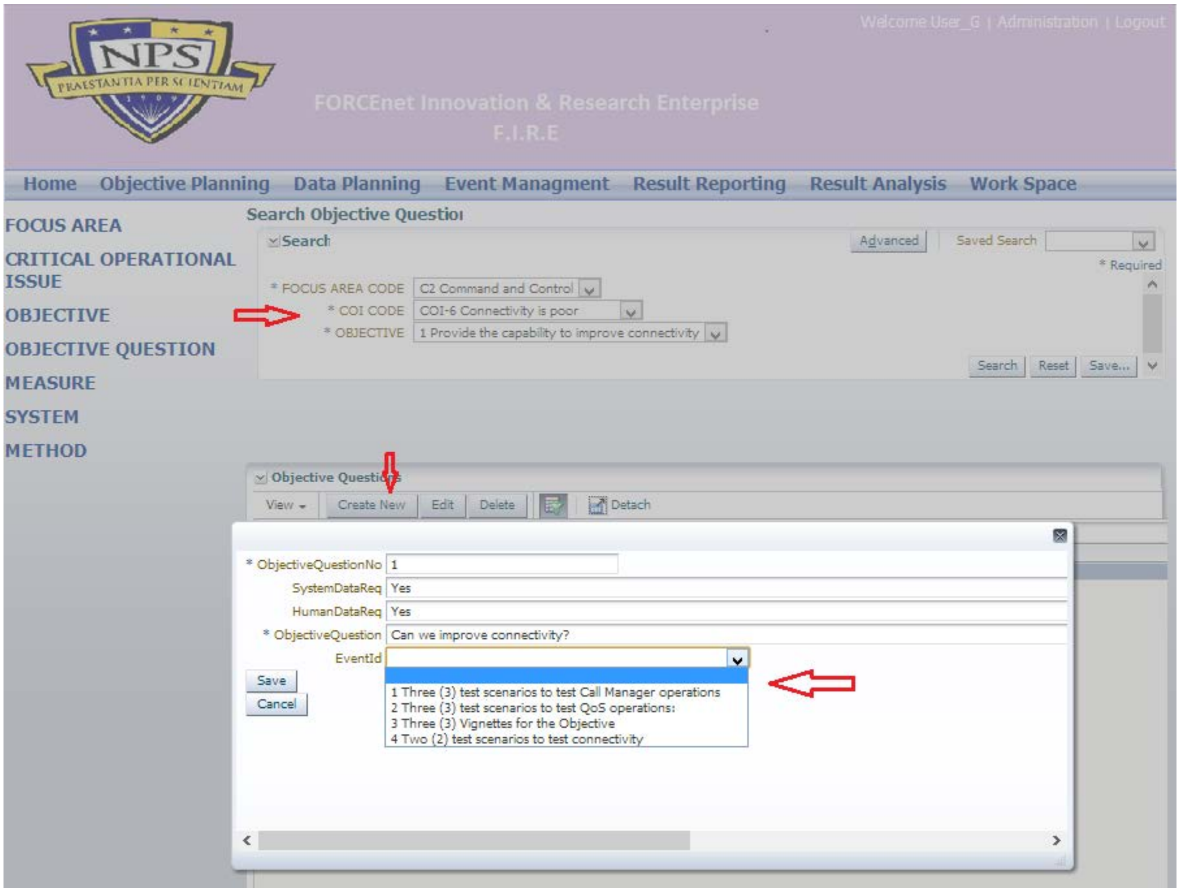
TechDescription: Provide the capability to improve connectivity

FocusAreaCode: C2

CoiCode: COI-6

Figure 27. A snapshot of the popup window to create a new Objective.

- **Step 2:** A member of the Data Control group creates a new Objective-Question for the Objective that has been created in step 1. After clicking on the Objective Question link under the Objective Planning page, the user selects focus area C2, COI-6, and Objective number 1 from the dropdown menu, then, clicks on the CreateNew button (Figure 28). A popup window appears containing a blank form with the Objective Question attribute fields, where the user can fill in the data (e.g., ObjectiveQuestionNo: 1), and then clicks on the Save button. Note that the event field is a dropdown menu where the user can choose to assign the objective question to an event or leave it blank and edit it later. The user also has the option to cancel the operation by simply clicking on the Cancel button.



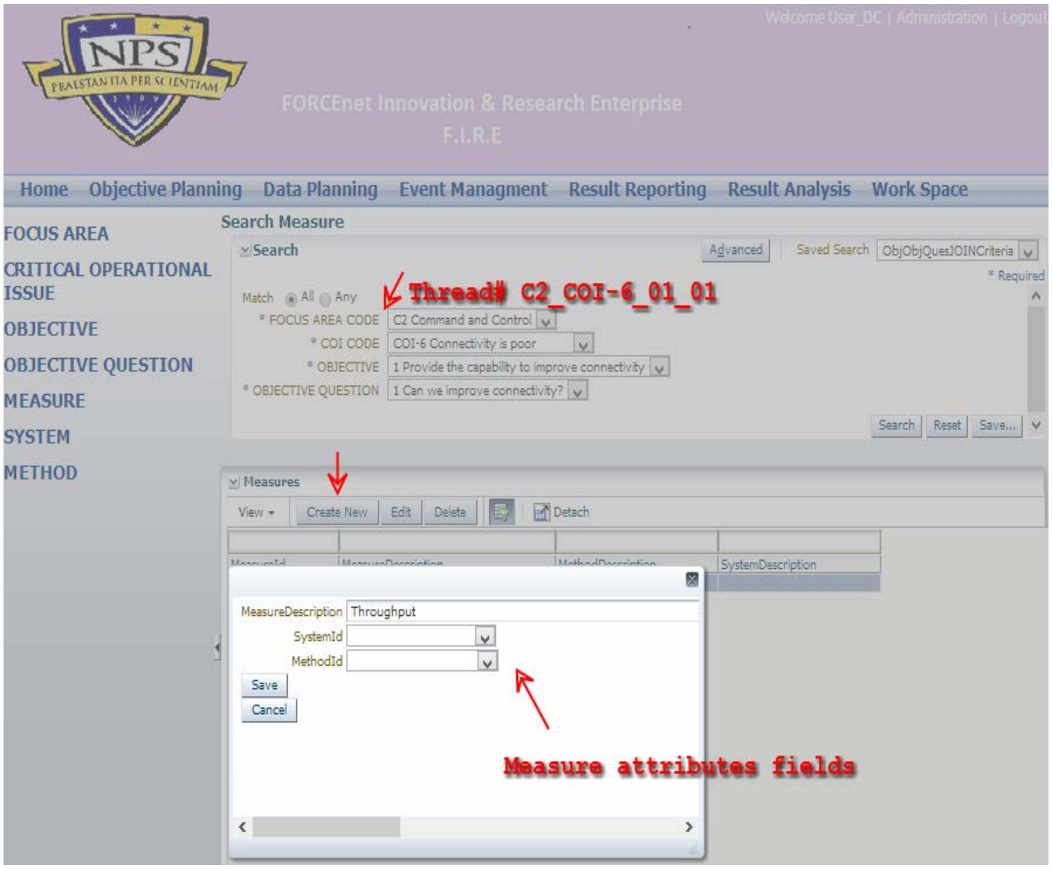
* FOCUS AREA CODE: C2 Command and Control
* COI CODE: COI-6 Connectivity is poor
* OBJECTIVE: 1 Provide the capability to improve connectivity

* ObjectiveQuestionNo: 1
SystemDataReq: Yes
HumanDataReq: Yes
* ObjectiveQuestion: Can we improve connectivity?
EventId:

1 Three (3) test scenarios to test Call Manager operations
2 Three (3) test scenarios to test QoS operations:
3 Three (3) Vignettes for the Objective
4 Two (2) test scenarios to test connectivity

Figure 28. Figure 28: A snapshot of the popup window to create a new Objective-Question

- **Step 3:** A member of the Data Control group creates a new Measure for the Objective-Question that has been created in step 2. After clicking on the Measure link under the Objective Planning page, the user selects focus area C2, COI-6, Objective number 1 and Objective Question number 1 (this represents the thread# C2_COI-6_01_01) from the dropdown menu, then clicks on the CreateNew button (Figure 29). A popup window appears containing a blank form of the Measure attribute fields where the user can fill in the data (e.g., Measure Description: Throughput), and then clicks on the Save button. The System and Method fields are dropdown menus where the user can choose to assign the required System and Method, or leave them empty until the user determines the appropriate system and method for that measure. The user also has the option to cancel the operation by simply clicking on the Cancel button.



Saved Search: ObjObjQuesJOINCriteria

Match ● All ○ Any

* FOCUS AREA CODE: C2 Command and Control

* COI CODE: COI-6 Connectivity is poor

* OBJECTIVE: 1 Provide the capability to improve connectivity

* OBJECTIVE QUESTION: 1 Can we improve connectivity?

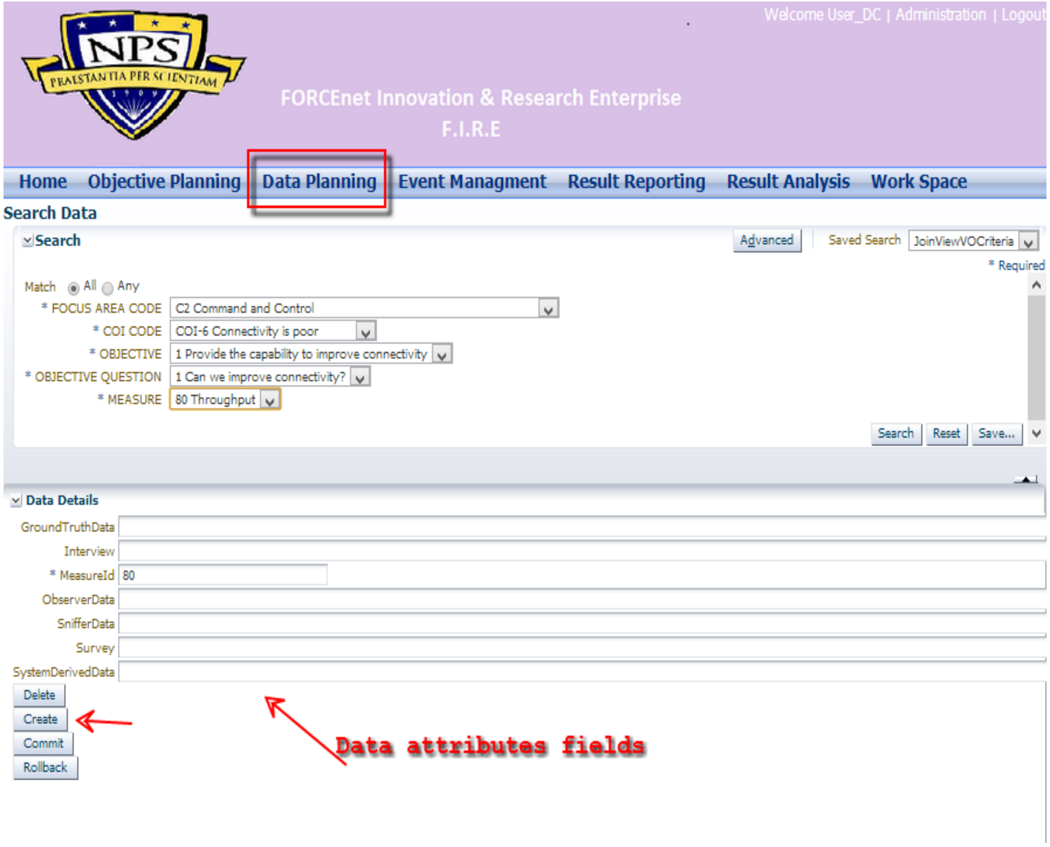
MeasureDescription: Throughput

SystemId:

MethodId:

Figure 29. A snapshot of the popup window to create a new Measure

- **Step 4:** A member of the Data Control group creates a new Data record required for the Measure that has been created in step 3. Since specifying the required Data is part of the Data Planning phase, the user clicks first on the Data Planning link in the main navigation bar (Figure 30). The user then searches the thread number and the required measure, and creates new data for that measure by filling in the data fields of the displayed blank form.



Saved Search: JoinViewVOCriteria

Match ● All ○ Any

* FOCUS AREA CODE: C2 Command and Control

* COI CODE: COI-6 Connectivity is poor

* OBJECTIVE: 1 Provide the capability to improve connectivity

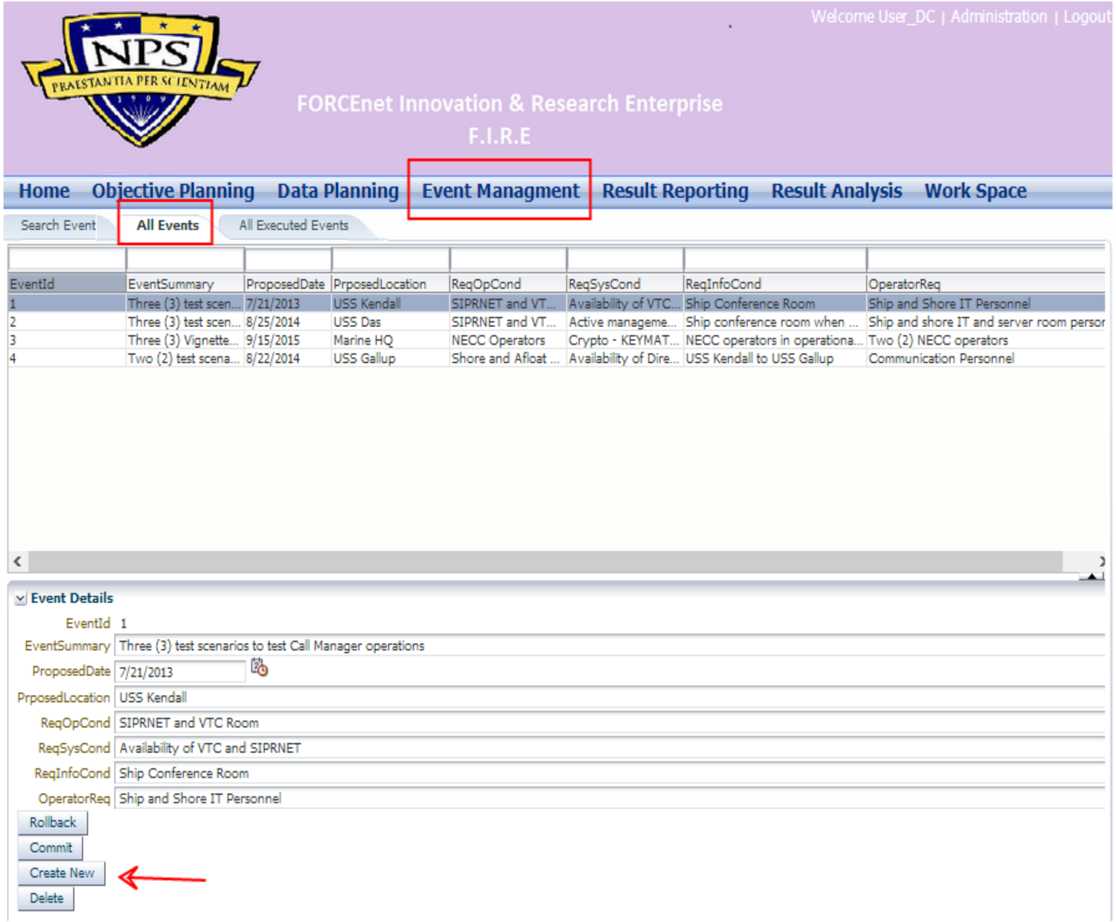
* OBJECTIVE QUESTION: 1 Can we improve connectivity?

* MEASURE: 80 Throughput

* MeasureId: 80

Figure 30. A snapshot of the Data Planning page to create a new Data

- **Step 5:** A member of the Data Control group creates a new Event to test one or several experiment Objective questions. The user clicks first on the Event Management page link in the main navigation bar and then selects the “All Events” tab (Figure 31). The user then clicks on the Create button to add new event information.



EventId	EventSummary	ProposedDate	PrposedLocation	ReqOpCond	ReqSysCond	ReqInfoCond	OperatorReq
1	Three (3) test scen...	7/21/2013	USS Kendall	SIPRNET and VT...	Availability of VTC...	Ship Conference Room	Ship and Shore IT Personnel
2	Three (3) test scen...	8/25/2014	USS Das	SIPRNET and VT...	Active manageme...	Ship conference room when ...	Ship and shore IT and server room persor
3	Three (3) Vignette...	9/15/2015	Marine HQ	NECC Operators	Crypto - KEYMAT...	NECC operators in operationa...	Two (2) NECC operators
4	Two (2) test scena...	8/22/2014	USS Gallup	Shore and Afloat ...	Availability of Dire...	USS Kendall to USS Gallup	Communication Personnel

EventId 1

EventSummary Three (3) test scenarios to test Call Manager operations

ProposedDate 7/21/2013

PrposedLocation USS Kendall

ReqOpCond SIPRNET and VTC Room

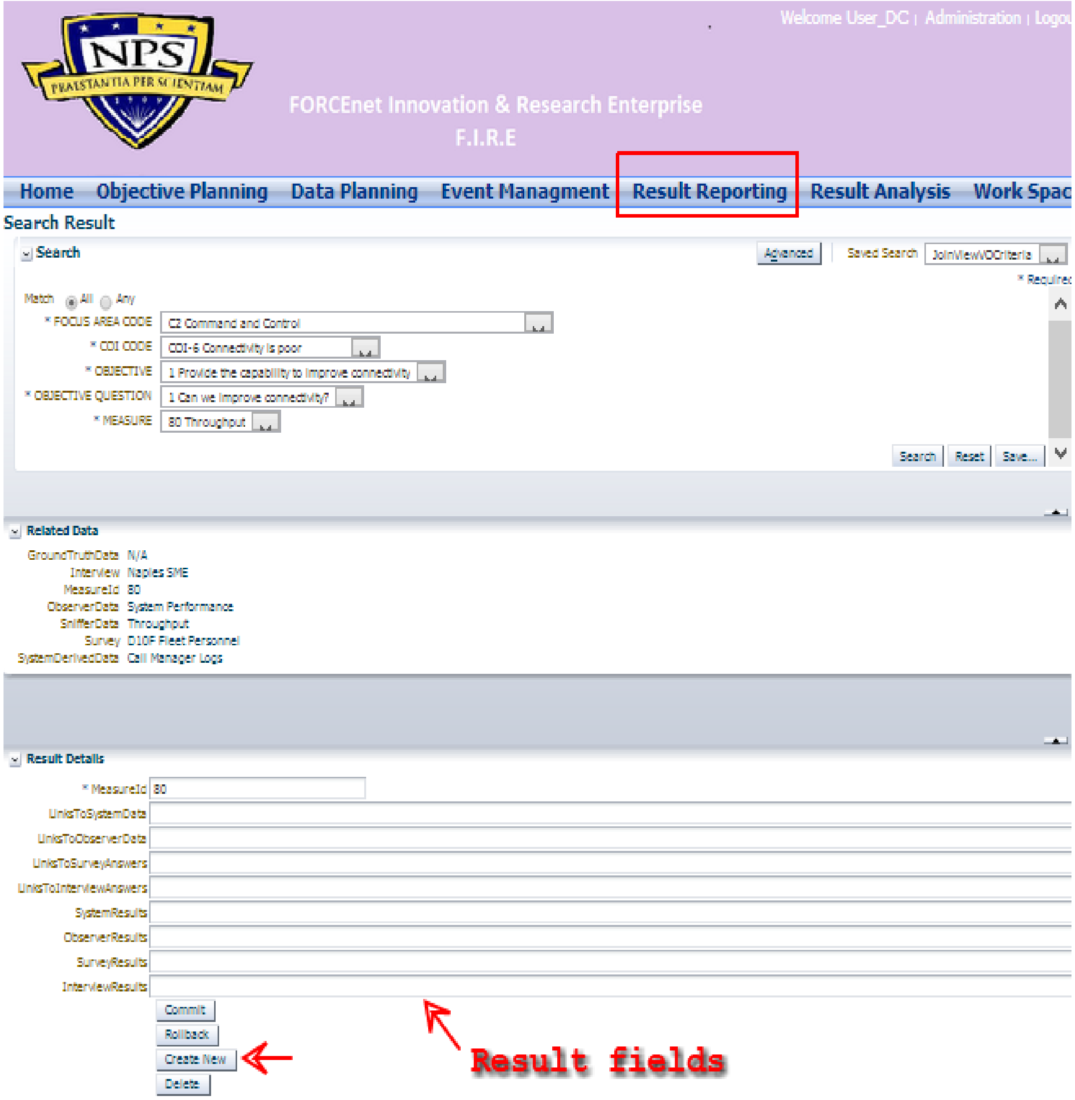
ReqSysCond Availability of VTC and SIPRNET

ReqInfoCond Ship Conference Room

OperatorReq Ship and Shore IT Personnel

Figure 31. A snapshot of the Event Management page to create a new Event

- **Step 6:** A member of the Data Control group creates a new Result following an event experiment where data has been collected on a specific measure, and is now ready to be entered into the system. The user clicks on the Result Reporting phase link on the main navigation bar, and then clicks on the Create button after searching for a measure using an appropriate thread number (Figure 32). A blank form is displayed that allows the user to fill in the data results for that measure.



Search Result

Search

Saved Search: JoinViewOCriteria

* Required

Match: ● All ○ Any

* FOCUS AREA CODE	C2 Command and Control
* COI CODE	COI-6 Connectivity is poor
* OBJECTIVE	1 Provide the capability to improve connectivity
* OBJECTIVE QUESTION	1 Can we improve connectivity?
* MEASURE	80 Throughput

Related Data

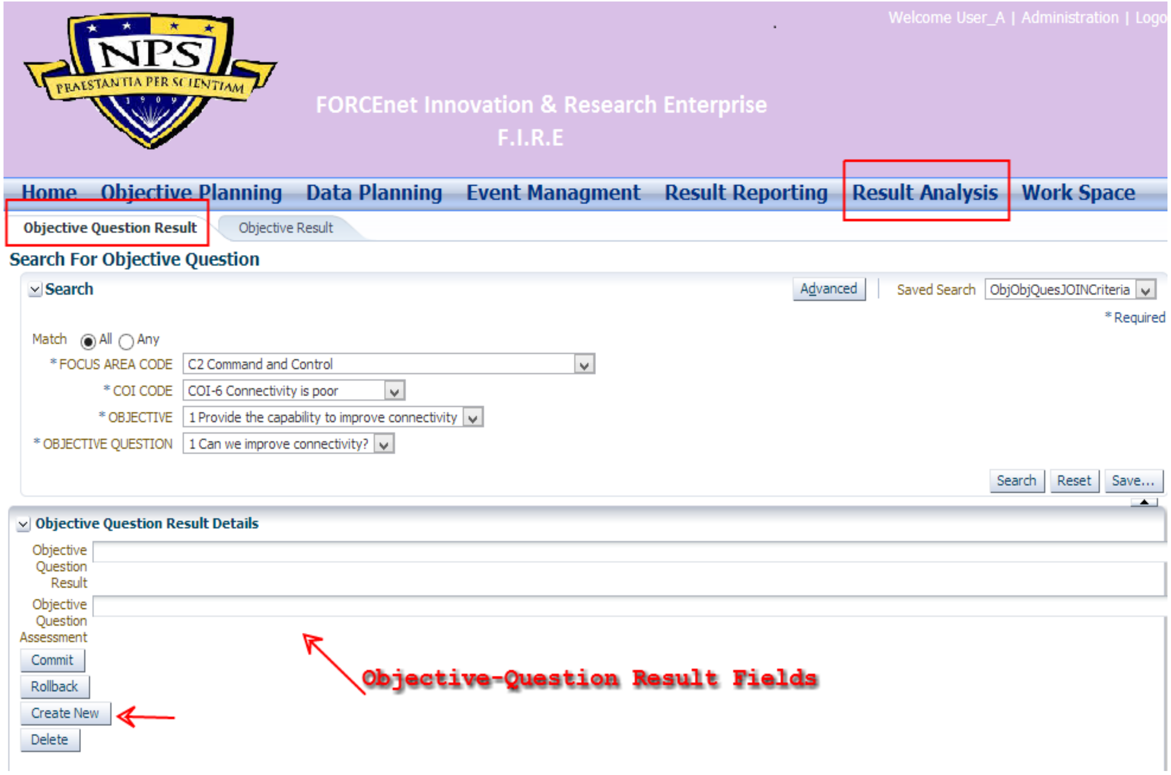
GroundTruthData	N/A
Interview	Naples SME
MeasureId	80
ObserverData	System Performance
SnifferData	Throughput
Survey	D10F Fleet Personnel
SystemDerivedData	Call Manager Logs

Result Details

* MeasureId	80
LinksToSystemData	
LinksToObserverData	
LinksToSurveyAnswers	
LinksToInterviewAnswers	
SystemResults	
ObserverResults	
SurveyResults	
InterviewResults	

Figure 32. A snapshot of the Result Reporting page to create a new Result

- **Step 7: A** member of the Analysis group creates a new Objective Question Result. That team member basically assesses the data collected for the Objective Question (created in step 2) and then inputs a result. This is done by first clicking on the Result Analysis page link on the main navigation bar, and then clicking on the Create button after searching the appropriate thread number and measure (Figure 33).


Welcome User_A | Administration | Logo
NPS PRAESTANTIA PER SCIENTIAM 1909
FORCEnet Innovation & Research Enterprise
F.I.R.E
Home
Objective Planning
Data Planning
Event Managment
Result Reporting
Result Analysis
Work Space
Objective Question Result
Objective Result
Search For Objective Question
Search
Advanced
Saved Search
* Required
Search
Reset
Save...
Objective Question Result Details
Commit
Rollback
Create New
Delete
Objective-Question Result Fields

Saved Search: ObjObjQuesJOINCriteria

Match ● All ○ Any

* FOCUS AREA CODE: C2 Command and Control

* COI CODE: COI-6 Connectivity is poor

* OBJECTIVE: 1 Provide the capability to improve connectivity

* OBJECTIVE QUESTION: 1 Can we improve connectivity?

Objective Question Result:

Objective Question Assessment:

Figure 33. A snapshot of the Result Analysis page to create a new Objective-Question Result

- **Step 8:** a member of the Analysis group creates a new Objective Result. This is done in a similar way as step 7.

D. SUMMARY

In this chapter, we provided a brief description of the application development process. We also provided a detailed description of the application functionalities and features. By providing a scenario that covers the five phase of the experiment coordination, the reader should now have a clear idea about the functions that *i*FIRE could do. In Chapter VII, we will discuss the test and evaluation of the application.

VI. TEST AND EVALUATION

A. INTRODUCTION

In this chapter, we discuss the testing and evaluation of the *i*FIRE prototype application we developed. The aim of the testing is to make sure that the application satisfies the predefined set of functions, the required level of performance and quality, and ease of use. The process of test and evaluation (T&E) should continue throughout the life cycle of the application and future development phases. Questions like “did we build it right?” and “did we build the right thing?” should be repeatedly asked after each major development phase of *i*FIRE.

The T&E task has been carried out in a standalone and demonstration mode; the application has been tested running on a single computer in a development environment with all the necessary servers configured. Due to the limited time available for this research, real-time testing and evaluation in a production environment with concurrent multi users was not possible. The aim of this T&E is to first ensure that the application runs properly without issues, and second to confirm that it does what it is intended to do. Taking into consideration the scope and time of this research, the best option to conduct this T&E was to invite as many members of FIRE stakeholders, run the system before them, demonstrate all its functions and features, and collect their feedback and recommendations. There were eight participants only in this T&E operation. It is important to point out that this is an informal evaluation and that the evaluators of the prototype may not accurately represent the users of the actual FIRE System. The purpose of this T&E is to get a preliminary assessment of *i*FIRE.

The remainder of this chapter describes the T&E methodology and discusses the FIRE stakeholder’s feedback.

B. TEST AND EVALUATION METHODOLOGY

A presentation of the prototype application was arranged to evaluate application functionalities, usability and effectiveness. All attendees were users of the actual FIRE system with different responsibilities and roles such as system administrators, experiment developers, experiment participants and result analysts. Before the demonstration, all attendees were handed an *i*FIRE test and evaluation questionnaire and were asked to complete it at the conclusion of the demonstration. The questionnaire was designed to evaluate the following two distinct areas: usability and functionality, along with a general feedback. For both the usability and functionality areas sections, the attendees were presented with several statements and asked to provide their agreement or disagreement to these statements along the following scale:

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

In the following sections we describe the T&E questionnaire and then conduct an analysis of the feedback collected from the participants in this T&E effort.

1. Usability

In this part, we wanted to make sure that the *i*FIRE is easy to use and that a user will be able to perform the required tasks with effectiveness, efficiency, and satisfaction. We asked the attendees to consider and select their level of agreement with the following statements:

- It is easy to navigate through this web-based application
- It is easy to find the information I want
- It is easy to use this Web-based application upon my first visit
- Navigating the application screen seems intuitive

- I like the application UI: colors, layout, navigation bar, links, and tables
- The application offers powerful Search functionality
- The implementation of dropdown menu option simplifies “create and update” operations
- It is easy to upload and download document to/from the Application

2. Functionality

In this section, we want to make sure that the system will support all the use cases and all functional requirements for the following experimental phases: Objective Planning, Data Planning, Event Management, Result Reporting, and Result Analysis. We asked attendees to consider and select their level of agreement with the following statements:

- All use cases are supported
- The application addresses the functions needed for the Objective planning phase
- The application address the functions needed for the Data planning phase
- The application address the functions needed for the Events Management
- The application address the functions needed for the Result Reporting phase
- The application address the functions needed for the Result Analysis phase

3. General Feedback

The purpose of this section is to obtain additional comments from the attendees about the overall system, areas that need improvement in future development, and any changes or additional features that should be added to the system. Attendees were asked the following questions:

- Overall, what is your opinion about this application?
- What is it about this application that you would most like to see improved?
- What changes or additional features would you suggest for this application?

C. RESULTS ANALYSIS

A summary of the user's feedback is presented and discussed in the following sections.

1. Usability

Attendee's answers to the system usability statement are:

- 100% of participants agreed that it was easy to navigate through this Web-based application.
- 100% of participants agreed that it was easy to find the information they want.
- 83.33% of participants agreed that it was easy to use this Web-based application upon their first visit. While the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 100% of participants agreed that navigating the application screen seemed intuitive.
- 68 % of participants agreed that they liked the application UI: colors, layout, navigation bar, links, and tables. Only 16 % disagreed, while the remaining 16% were neutral.
- 83.33% of participants agreed that the application offered powerful search functionality, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 100% of participants agreed that the implementation of dropdown menu options simplified "create and update" operations.
- 100% of participants agreed that it was easy to upload and download documents to/from the application.

2. Functionalities

Attendee's answers to the functional statement included the following:

- 83.33% of participants agreed that all use cases were supported, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 83.33% of participants agreed that the application addressed the functions needed for the Objective planning phase, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.

- 83.33% of participants agreed that the application address the functions needed for the Data planning phase, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 83.33% of participants agreed that the application address the functions needed for the Events Management, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 83.33% of participants agreed that the application addressed the functions needed for the Result Reporting phase, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.
- 83.33% of participants agreed that the application addresses the functions needed for the Result Analysis phase, while the remaining 16.67% were neutral. There was no disagreement by any participants to this question.

It may be relevant to highlight that the 16.67% participant who remained neutral did not have a strong familiarity regarding the functionality of the FIRE system. So it can be inferred that most participants who were familiar with the FIRE system functionality supported this application's functionalities.

3. General Feedback

Attendees provided comments to the following questions:

a. Overall, what is your opinion about this application?

Surveyed users (83.3%) expressed positive feedback about the application, while 16.67 % did not respond. Users added the following comments to this question:

- "This is excellent and will be the model for the next version of FIRE"
- "Runs well. Fast"
- "Great"
- "Looks highly functional. Provides good functionality for multi-level data manipulation"
- "Good approach but oracle centric"

b. What is it about this application that you would most like to see improved?

Attendees answered this question with interesting comments that would help set the direction for future system development areas. Users added the following comments to this question:

- “The normalization of the data mode sorting the information much easier and should really reduce development time to construct new experiments”
- “Design, look ”
- “Work flow”
- “Reporting options”
- “I would like to see better integration between the different components. Spaces, portal, content”

c. What changes or additional features would you suggest for this application?

Answers to this question will also be valuable for future development. Users added the following comments to this question:

- “Custom query capabilities could be added”
- “An easy way to link files without copy/pasting URLs”
- “Ability for users to not have to click through all LOVs to get to their applicable data set each time”
- “Security”

D. SUMMARY

Answers and feedback on the T&E questionnaire has shown that *i*FIRE application is a significant step up toward improving and supporting the coordination of large-scale experimentation. The majority of the attendees found the application easy to use, that it supports all the experiment use case, and satisfies all the functions needed for the different experiment phases. Feedback also indicated that additional features should be considered in future development such as: better design, workflow, and reporting capability; enhanced security; customized queries; and better integration of application with collaborative tools.

VII. CONCLUSION

This chapter summarizes the effort of this thesis for the analysis, design, and implementation of the proof-of-concept prototype. Lessons learned are then drawn regarding the implementation of this prototype to benefit the development of future FIRE prototypes/system. The chapter also presents directions for future research opportunities.

This chapter is organized as follows: Section A summarizes the thesis work, Section B discusses lessons learned, and Section C proposes future research recommendations.

A. SUMMARY

In this thesis, we discussed the need for conducting large-scale experimentation to test new capabilities necessary to achieve FORCEnet concept. We also discussed the role the actual FIRE system is playing to support the experiments coordination as well as the collaboration of the different players involved in those experiments. Although the current FIRE provided great functionality, according to users' feedback, it still lacks many features. We believe that iFIRE, the prototype application we designed and built as the goal of this research, has implemented a simple representation of the experiment phases as well as many essential features, such as powerful search capabilities. It adds considerable improvement to the actual FIRE and may be considered the first step toward the next generation of FIRE.

The requirements analysis was divided into two main parts: the data model and the process model requirements. The data model was designed to tightly reflect the five phases of the experiment coordination: Objective Planning, Data Planning, Event Management, Result Reporting, and Result Analysis. The process model defined the actors in these phases: Guidance Group, Data Control Group, Analysis Group, and Admin Group. We then defined the use cases that represent the functions that could be performed by actors in each phase.

After the requirements validation, we developed iFIRE using MVC as the design pattern and Oracle ADF, which implements MVC as the development framework. Many Oracle tools were used in the application development. Oracle SQL Developer Data Modeler and Oracle SQL Developer were used for the database modeling and implementation. Oracle JDeveloper is the main Integrated Development Environment (IDE), and it was used to develop and deploy (using the integrated WebLogic server) the application for testing. Oracle WebCenter Framework provided the customization at run-time capability. Oracle WebCenter: Spaces provided the prebuilt spaces, including many built-in collaborative tools such as messages, events, and lists. Finally, Oracle content server provided content management capability.

The user interface of the application was designed to reflect the five phase of the experiment coordination. In addition, the page layout and colors, the navigation links, and buttons were designed and used to achieve a high level of usability. The search feature was widely implemented to simplify many of the functions required in the different phases of the experiments. The application usability and functionality were evaluated by users who play different roles in the current FIRE system. We ran and demonstrated iFIRE to those users and asked them to complete a detailed questionnaire to evaluate the prototype application usability and functionality. User's feedback indicated that all use cases were implemented and that the application achieved a high level of usability.

B. LESSONS LEARNED

This section summarizes some of the lessons learned during the effort of the thesis that may be helpful for any future research. The following is an outline of these lessons learned:

- • Rapid prototyping provided the systematic narrowing down of user's requirements and focused development efforts. In fact, implementing some use cases at early stage in this research provided us with a better understanding of the users' expectations.
- • Adequate security clearance and full access privilege to existing data and actual system would lead to better understanding of the users' requirements and the actual FIRE system shortcomings.
- • Despite the number of books available to learn Oracle development tools, materials available online including Oracle website, discussion forum, blogs, and YouTube videos should be considered first, at least until an intermediate level of competency is reached. These means are very helpful to understand the tools capabilities and for development problem solving.
- • Getting some Oracle training courses was very helpful to accelerate the development of the tools learning process, and it provided an opportunity to meet many subject matter experts and ask relevant questions.
- • JDeveloper provides a visual and declarative development approach; however, a minimum knowledge of Java would help developers integrate certain system requirements and to overcome any limitations of basic tools functionality.
- • Building our application on a normalized database (which should be the case for a well-designed database) provides many benefits such as: eliminating redundancy and modification anomalies, accurately representing the user view of data, allowing reporting and analysis at lower level of granularity, and simplifying the modification of the data model
- • Security is a very important aspect of deployment. Therefore, it should be considered carefully during development , mastered and implemented correctly.
- • Powerful computers (at least 6GHz in RAM and high processor) and large display (at least 27" display) would significantly increase the development efficiency.

C. FUTURE AREAS OF RESEARCH

In our research, we conducted requirements analysis and developed a proof-of-concept prototype that could be the first step to build the next generation of FIRE. This work could be extended to many interesting areas. The four primary areas that we recommend are deployment, requirement analysis, security, and services (collaboration tools) integration options. These suggested research areas are described below.

1. *i*FIRE Deployment

*i*FIRE deployment was beyond the scope and time of this thesis. An interesting research area would to be to actually deploy the application in a controlled environment to capture user feedback on application functionality and usability. User feedback should then be used to influence the modification and upgrade of follow on prototypes.

2. Further Requirements' Analysis

Large-scale experimentation involves wide range of experiment categories. A promising research area would be to conduct a very comprehensive requirements analysis involving the redesign of the experiment planning and structure. The requirements analysis should heavily involve FIRE users and document their feedback, requirements, and concerns. The outcome should lead to the design of a data and process model that would constitute the heart of FIRE next generation system.

3. Security

A detailed analysis of FIRE security requirements is another promising research area. This could involve investigating Oracle security options to identify the optimal security policy for future generation of FIRE as well as reviewing DoD security policies for applications deployment.

4. Examining the Services Integration options

There are two options for integrating services like discussion forums, instant messaging, events, lists, wikis, and tags. The first option would be to build a portal that implement FIRE business logic (use cases) and then configure the necessary servers to integrate those services in it. The second option would be to use the prebuilt Oracle WebCenter Space, which contains those built-in services. Then, FIRE business logic should be developed and integrated in the Oracle WebCenter Space. A promising research area would be to examine both options and draw some recommendations and preferences.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DDL SCRIPT FOR *I*FIRE DATABASE

```
-- Generated by Oracle SQL Developer Data Modeler 3.3.0.747
--   at:        2013-09-01 16:16:36 PDT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g




CREATE TABLE COI
  (
    COI_Code        VARCHAR2 (10) NOT NULL ,
    COI_Description VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE COI ADD CONSTRAINT COI_PK PRIMARY KEY
(
  COI_Code
)
;

CREATE TABLE Data
  (
    Measure_id          INTEGER NOT NULL ,
    Sniffer_Data        VARCHAR2 (255) ,
    Ground_Truth_Data   VARCHAR2 (255) ,
    System_Derived_Data VARCHAR2 (255) ,
    Observer_Data       VARCHAR2 (255) ,
    Survey              VARCHAR2 (255) ,
    Interview           VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Data ADD CONSTRAINT Data_PK PRIMARY KEY
(
  Measure_id
)
;

CREATE TABLE Event
  (
    Event_id         INTEGER NOT NULL ,
    Event_Summary    VARCHAR2 (255) ,
    Proposed_Date    DATE ,
    Prposed_Location VARCHAR2 (255) ,
    Req_Op_Cond      VARCHAR2 (255) ,
    Req_Sys_Cond     VARCHAR2 (255) ,
    Req_Info_Cond    VARCHAR2 (255) ,
    Operator_Req     VARCHAR2 (255)
  )
  LOGGING ;
```

```
ALTER TABLE Event ADD CONSTRAINT Event_PK PRIMARY KEY
(
  Event_id
)
;

CREATE TABLE Executed_Event
  (
    Event_id                INTEGER NOT NULL ,
    Executed_Event_Summary  VARCHAR2 (255) ,
    Ops_Dev                 VARCHAR2 (255) ,
    Sys_Dev                 VARCHAR2 (255) ,
    Info_Dev                VARCHAR2 (255) ,
    Operator_Dev            VARCHAR2 (255) ,
    Context_Summary         VARCHAR2 (255) ,
    Impact_Results_Content  VARCHAR2 (255) ,
    Impact_Results_Validity VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Executed_Event ADD CONSTRAINT Executed_Event_PK PRIMARY
KEY
(
  Event_id
)
;

CREATE TABLE Focus_Area
  (
    Focus_Area_Code        VARCHAR2 (10) NOT NULL ,
    Focus_Area_Description VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Focus_Area ADD CONSTRAINT Focus_Area_PK PRIMARY KEY
(
  Focus_Area_Code
)
;

CREATE TABLE Measure
  (
    Measure_id          INTEGER NOT NULL ,
    Measure_Description VARCHAR2 (255) ,
    Obj_Quest_id        INTEGER NOT NULL ,
    System_id           INTEGER ,
    Method_id           INTEGER
  )
  LOGGING ;
ALTER TABLE Measure ADD CONSTRAINT Measure_PK PRIMARY KEY
(
  Measure_id
)
;
```

```
CREATE TABLE Method
  (
    Method_id          INTEGER NOT NULL ,
    Method_Description VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Method ADD CONSTRAINT Method_PK PRIMARY KEY
(
  Method_id
)
;

CREATE TABLE Obj_Que_Res
  (
    Obj_Quest_id        INTEGER NOT NULL ,
    Obj_Ques_Result     VARCHAR2 (255) ,
    Obj_Ques_Assessment VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Obj_Que_Res ADD CONSTRAINT Obj_Que_Res_PK PRIMARY KEY
(
  Obj_Quest_id
)
;

CREATE TABLE Objective
  (
    Objective_id     INTEGER NOT NULL ,
    Objective_No     INTEGER NOT NULL ,
    Tech_Description VARCHAR2 (255) ,
    Focus_Area_Code  VARCHAR2 (10) NOT NULL ,
    COI_Code         VARCHAR2 (10) NOT NULL
  )
  LOGGING ;
ALTER TABLE Objective ADD CONSTRAINT Objective_PK PRIMARY KEY
(
  Objective_id
)
;

CREATE TABLE Objective_Analysis
  (
    Objective_id            INTEGER NOT NULL ,
    POC_ID                  NUMBER NOT NULL ,
    Objective_Analysis_Task VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Objective_Analysis ADD CONSTRAINT Objective_Analysis_PK
PRIMARY KEY
(
  Objective_id, POC_ID
)
;
```

```
CREATE TABLE Objective_Question
  (
    Obj_Quest_id          INTEGER NOT NULL ,
    Objective_Question_No INTEGER NOT NULL ,
    System_Data_Req       VARCHAR2 (255) ,
    Human_Data_Req        VARCHAR2 (255) ,
    Objective_Question    VARCHAR2 (255) ,
    Objective_id          INTEGER NOT NULL ,
    Event_id              INTEGER
  )
  LOGGING ;
ALTER TABLE Objective_Question ADD CONSTRAINT Objective_Question_PK
PRIMARY KEY
(
  Obj_Quest_id
)
;

CREATE TABLE Objective_Result
  (
    Objective_id          INTEGER NOT NULL ,
    Objective_Result      VARCHAR2 (255) ,
    Obj_Result_Assessment VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Objective_Result ADD CONSTRAINT Objective_Result_PK
PRIMARY KEY
(
  Objective_id
)
;

CREATE TABLE POC
  (
    POC_ID       NUMBER NOT NULL ,
    First_Name   VARCHAR2 (25) ,
    Last_Name    VARCHAR2 (25) ,
    Organization VARCHAR2 (25) ,
    Email        VARCHAR2 (50) ,
    Phone        VARCHAR2 (15)
  )
  LOGGING ;
ALTER TABLE POC ADD CONSTRAINT POC_PK PRIMARY KEY
(
  POC_ID
)
;

CREATE TABLE Results
  (
    Measure_id                 INTEGER NOT NULL ,
    Links_to_System_Data       VARCHAR2 (255) ,
    Links_to_Observer_Data     VARCHAR2 (255) ,
    Links_to_Survey_Answers    VARCHAR2 (255) ,
    Links_to_Interview_Answers VARCHAR2 (255) ,
    System_Results             VARCHAR2 (255) ,
    Observer_Results           VARCHAR2 (255) ,
    Survey_Results             VARCHAR2 (255) ,
    Interview_Results          VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE Results ADD CONSTRAINT Results_PK PRIMARY KEY
(
  Measure_id
)
;

CREATE TABLE System
  (
    System_id          INTEGER NOT NULL ,
    System_Description VARCHAR2 (255)
  )
  LOGGING ;
ALTER TABLE System ADD CONSTRAINT System_PK PRIMARY KEY
(
  System_id
)
;

ALTER TABLE Data ADD CONSTRAINT Data_Measure_FK FOREIGN KEY (
Measure_id ) REFERENCES Measure ( Measure_id ) NOT DEFERRABLE ;

ALTER TABLE Executed_Event ADD CONSTRAINT Executed_Event_Event_FK
FOREIGN KEY ( Event_id ) REFERENCES Event ( Event_id ) NOT
DEFERRABLE ;

ALTER TABLE Measure ADD CONSTRAINT Measure_Method_FK FOREIGN KEY (
Method_id ) REFERENCES Method ( Method_id ) NOT DEFERRABLE ;

ALTER TABLE Measure ADD CONSTRAINT Measure_Objective_Question_FK
FOREIGN KEY ( Obj_Quest_id ) REFERENCES Objective_Question (
Obj_Quest_id ) NOT DEFERRABLE ;

ALTER TABLE Measure ADD CONSTRAINT Measure_System_FK FOREIGN KEY (
System_id ) REFERENCES System ( System_id ) NOT DEFERRABLE ;

ALTER TABLE Obj_Que_Res ADD CONSTRAINT Obj_Que_FK_inObj_Que_Res
FOREIGN KEY ( Obj_Quest_id ) REFERENCES Objective_Question (
Obj_Quest_id ) NOT DEFERRABLE ;

ALTER TABLE Objective_Analysis ADD CONSTRAINT
Objective_Analysis_POC_FK FOREIGN KEY ( POC_ID ) REFERENCES POC (
POC_ID ) NOT DEFERRABLE ;
```

```
ALTER TABLE Objective ADD CONSTRAINT Objective_COI_FK FOREIGN KEY (
COI_Code ) REFERENCES COI ( COI_Code ) NOT DEFERRABLE ;

ALTER TABLE Objective_Analysis ADD CONSTRAINT
Objective_FK_in_Analysis_Obje FOREIGN KEY ( Objective_id )
REFERENCES Objective ( Objective_id ) NOT DEFERRABLE ;

ALTER TABLE Objective_Question ADD CONSTRAINT
Objective_FK_in_Obj_Que FOREIGN KEY ( Objective_id ) REFERENCES
Objective ( Objective_id ) NOT DEFERRABLE ;

ALTER TABLE Objective ADD CONSTRAINT Objective_Focus_Area_FK FOREIGN
KEY ( Focus_Area_Code ) REFERENCES Focus_Area ( Focus_Area_Code )
NOT DEFERRABLE ;

ALTER TABLE Objective_Question ADD CONSTRAINT
Objective_Question_Event_FK FOREIGN KEY ( Event_id ) REFERENCES
Event ( Event_id ) NOT DEFERRABLE ;

ALTER TABLE Objective_Result ADD CONSTRAINT
Objective_Result_Objective_FK FOREIGN KEY ( Objective_id )
REFERENCES Objective ( Objective_id ) NOT DEFERRABLE ;

ALTER TABLE Results ADD CONSTRAINT Results_Data_FK FOREIGN KEY (
Measure_id ) REFERENCES Data ( Measure_id ) NOT DEFERRABLE ;


-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                            15
-- CREATE INDEX                             0
-- ALTER TABLE                             29
-- CREATE VIEW                              0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           0
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0
-- CREATE ROLE                              0
-- CREATE ROLLBACK SEGMENT                  0
-- CREATE SEQUENCE                          0
-- CREATE MATERIALIZED VIEW                 0
-- CREATE SYNONYM                           0
-- CREATE TABLESPACE                                 0
-- CREATE USER                                       0
--
-- DROP TABLESPACE                                   0
-- DROP DATABASE                                     0
--
-- ERRORS                                            0
-- WARNINGS                                          0
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TASK FLOW SCREENSHOTS

Figures 34–39 are screenshots of the task flows implemented in *i*FIRE. These screenshots will help readers to see all the activities in each task flow. Moreover, they will be helpful to understand the control flows information, including : From Activity, From Outcome, and To Activity.

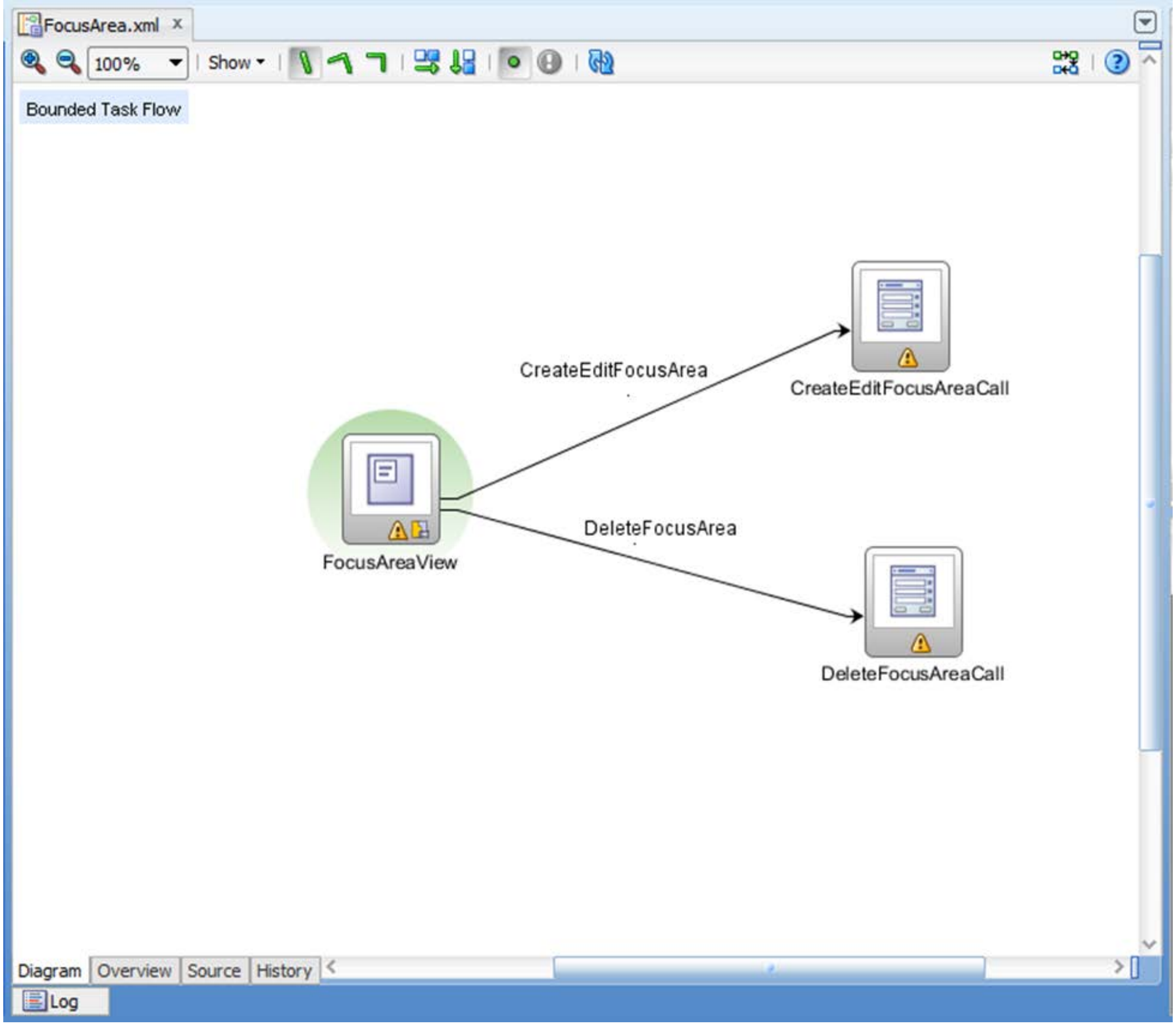


Figure 34. Focus Area Task Flow

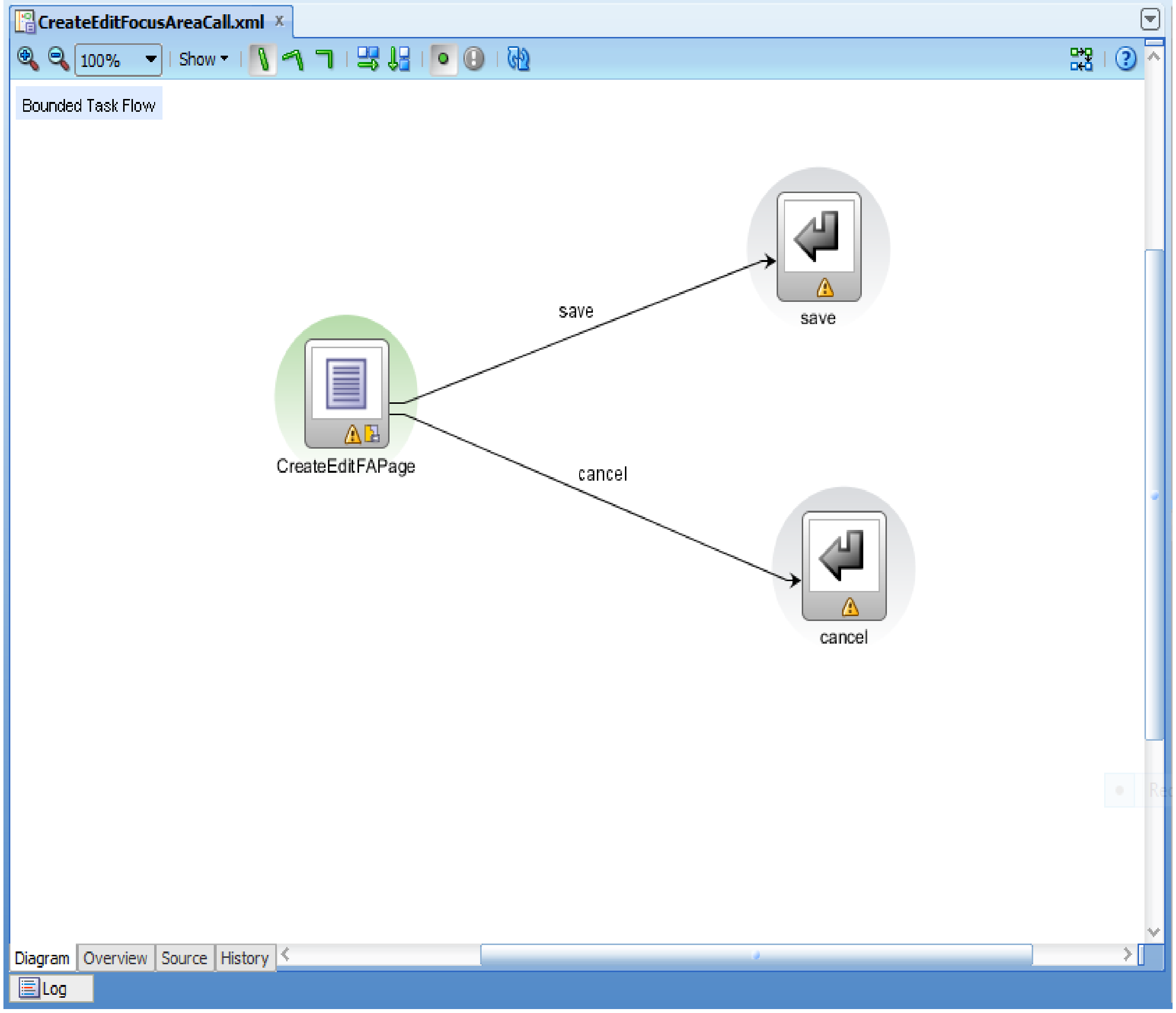


Figure 35. Create and Edit Focus Area Task Flow

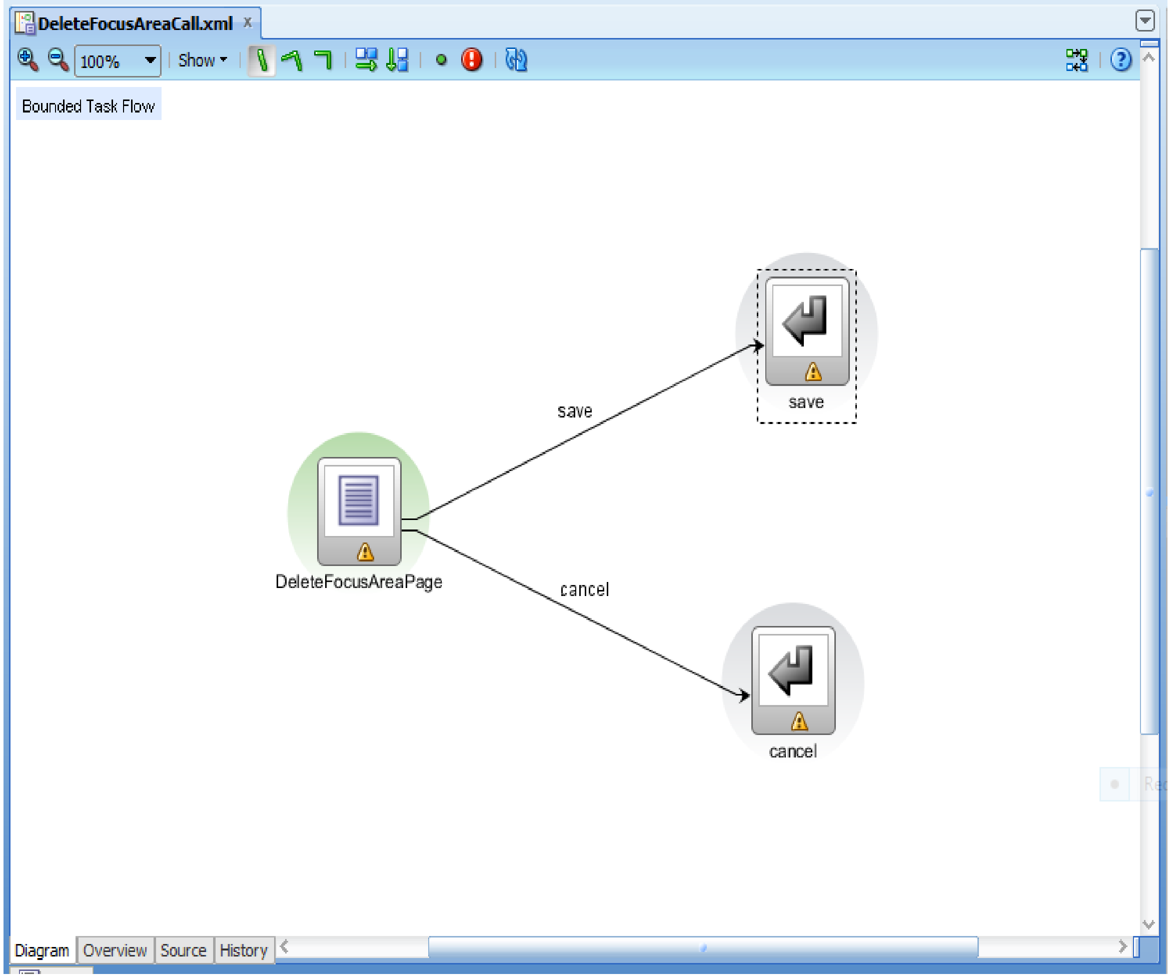


Figure 36. Delete Focus Area Task Flow

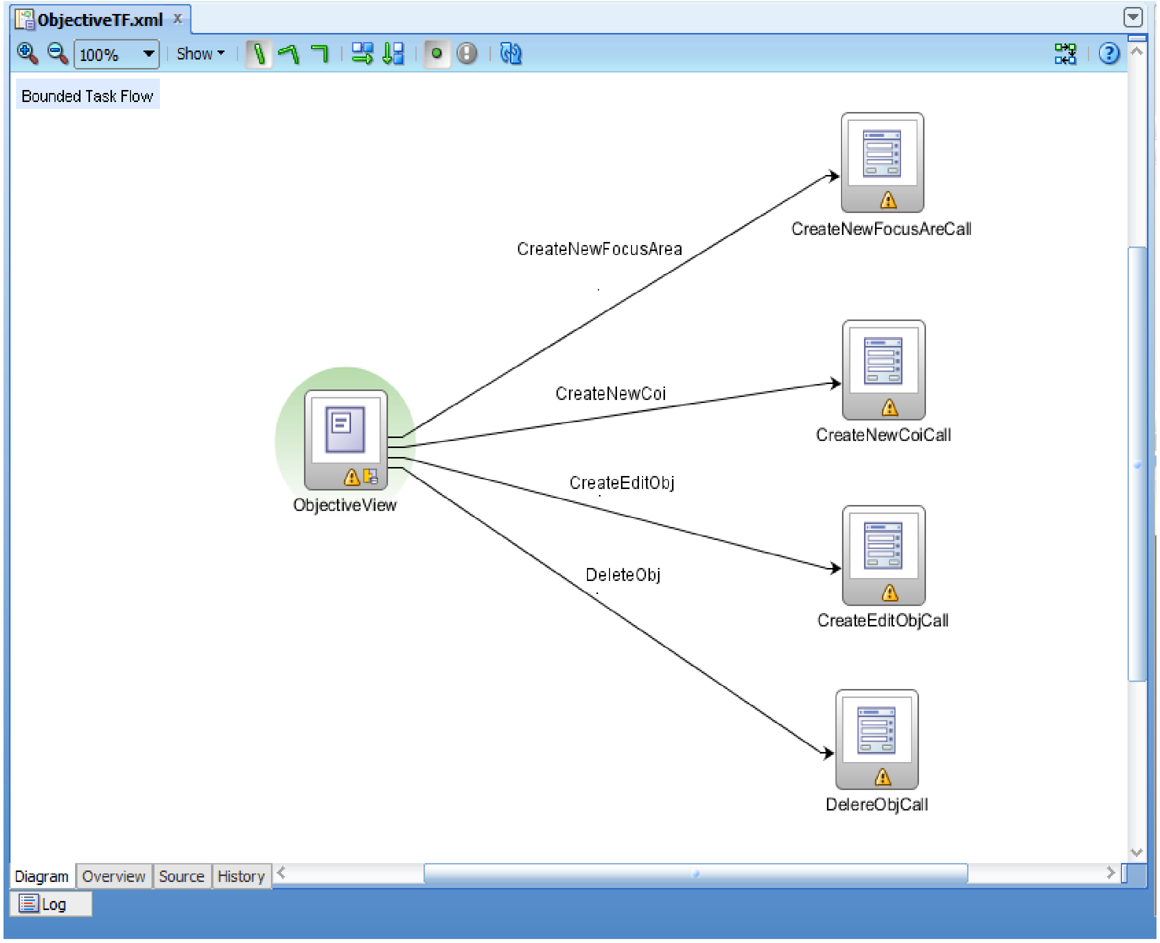


Figure 37. Objective Task Flow

□

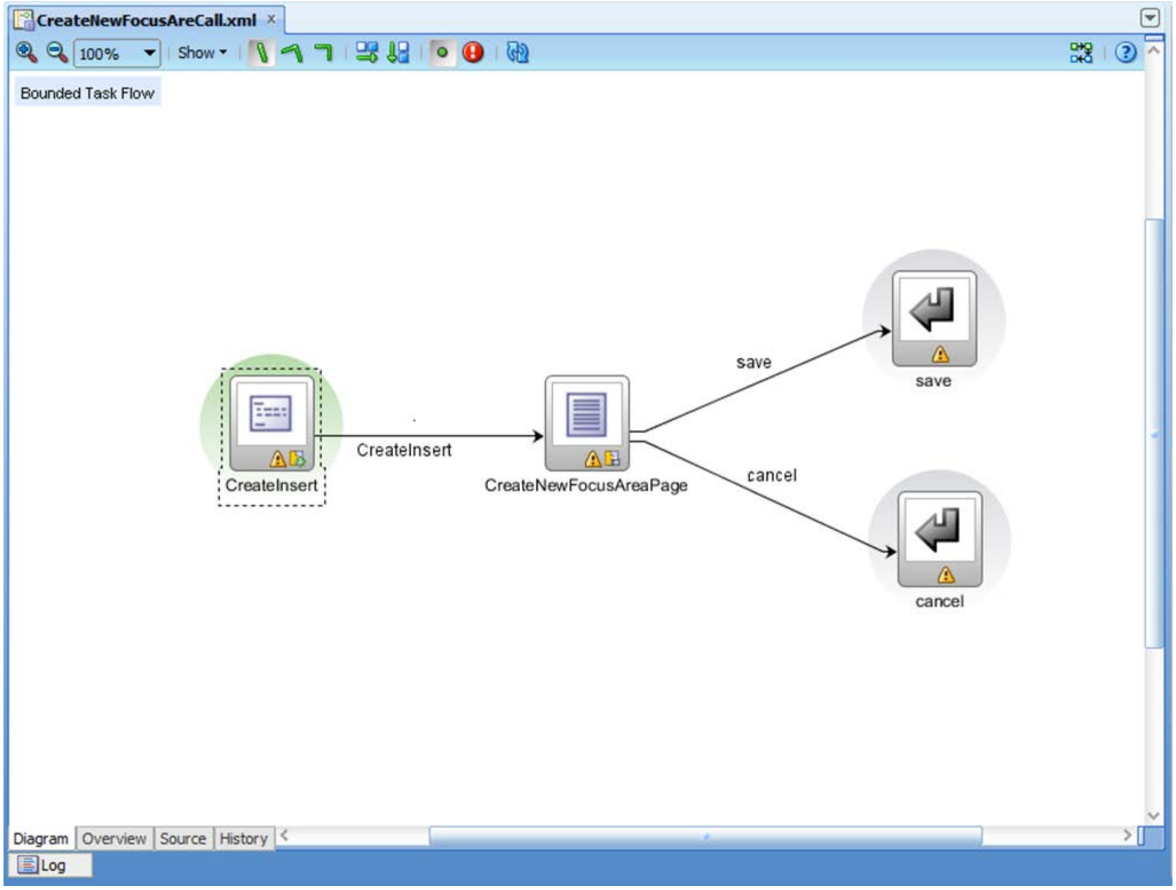


Figure 38. Create Focus Area Task Flow

□

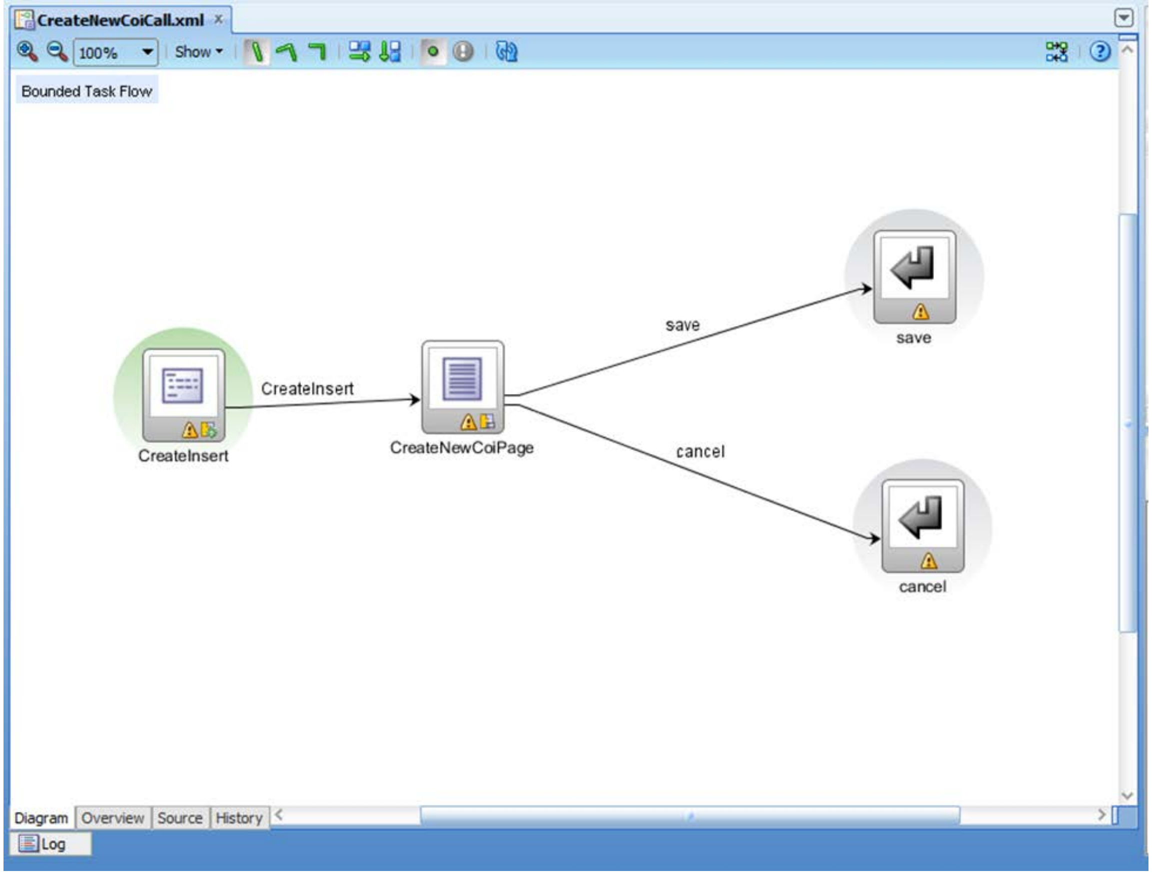


Figure 39. Create COI Task Flow

LIST OF REFERENCES

[1] *Future Naval Capabilities Guidebook*. Office of Naval Research. Arlington, VA. [Online]. Available: http://www.onr.navy.mil/Science-Technology/Directorates/Transition/~/media/Files/03T/FNC-Guidebook-2013.ashx

[2] B. J. McClain, and D. M. Smith, "Experimentation in a collaborative planning environment," M.S. Thesis, IS Department, Naval Postgraduate School, Monterey, CA, 2006. [Online]. Available: http://edocs.nps.edu/npspubs/scholarly/theses/2006/Jun/06Jun_McClain.pdf

[3] United States Naval Network Warfare Command, United States Marine Corps Combat Development Command, 2005. *FORCEnet A Functional Concept for the 21st Century*. Virginia Beach, VA:Naval Network Warfare Command. [Online]. Available: http://www.navy.mil/navydata/policy/forcenet/forcenet21.pdf.

[4] V. Clark. (2002, Oct.) "*Sea Power 21: Projecting Decisive Joint Capabilities*" [Online]. Available: http://www.navy.mil/navydata/cno/proceedings.html.

[5] "Interview with Vice Adm. James D. McArthur Jr.; Commander, Naval Network Warfare Command." *CHIPS Magazine.* October-December 2004. [Online]. Available: http://www.doncio.navy.mil/chips/ArticleDetails.aspx?ID=3273.

[6] Joint publication 3-13.1, 2012, [Online]. Available: http://www.airpower.au.af.mil/digital/Doctrine/du_jp3_13_1.pdf.

[7] K. Barrett, "The Trident Warrior experimentation process," M.S. Thesis, IS Department, Naval Postgraduate School, Monterey, CA, 2006, Available: http://edocs.nps.edu/npspubs/scholarly/theses/2005/Jun/05Jun_Barrett.pdf

[8] G. Schacher, "F.I.R.E. experiment planning and reporting structure," Naval Postgraduate School, Monterey, CA, Tech. Rep. NPS-IS-07-002, 2007.

[9] J. Satzinger, R. Jackson and S. Burd, *Systems Analysis and Design in a Changing*. Boston, MA: Cengage Learning, 2011.

[10] J. Purushothaman, *Oracle ADF Real World Developer's Guide.* Mumbai, India: Packt Publishing, 2012.

[11] L. Jamen*, Oracle Fusion Middleware Concepts Guide*. [Online]. Available*:* http://docs.oracle.com/cd/E17904_01/core.1111/e10103/intro.htm

[12] Oracle, “Oracle ADF 11g primer: Introduction to the building blocks of a fusion web application” [Online]. Available: http://www.oracle.com /technetwork/testcontent/oracle-11.pdf

[13] Andrew P. McAfee, “Enterprise 2.0: the dawn of emergent collaboration.” *MIT Sloan Management Review*, vol.47, no.3, spring 2006.

[14] Grant Ronald, *Quick Start Guide to Oracle Fusion Development: Oracle JDeveloper and Oracle ADF.* New York, NY: McGraw-Hill, 2011

[15] Oracle, “Oracle application development framework overview” [Online]. Available: http://www.oracle.com/technetwork/developer-tools/adf/adf-11-overview-1-129504.pdf

[16] R. Gordon et al. “Oracle fusion middleware fusion developer's guide for Oracle application development framework 11g” [Online]. Available: http://docs.oracle.com/cd/E16162_01/web.1112/e16182/toc.htm

[17] Oracle, “Oracle Jdeveloper” [Online]. Available: http://www.oracle.com /technetwork/developer-tools/jdev/jdeveloper11g-datasheet-1-133040.pdf

[18] S. Alapati , *Oracle WebLogic Server 11g Administration Handbook.*: New York, NY: McGraw-Hill Osborne, 2012.

[19] L. Jamen, *Oracle fusion middleware concepts guide*. [Online]. *Available*: http://docs.oracle.com/cd/E17904_01/core.1111/e10103/intro.htm.

[20] A. P. McAfee, “Enterprise 2.0: the dawn of emergent collaboration,” *MIT Sloan Management Review,* vol.47, no.3, spring 2006.

[21] Oracle, “Oracle webcenter suite 11g: The modern user experience platform for the enterprise and the Web” [Online]. Available: http://www .oracle.com/technetwork/middleware/webcenter/owcs-r11ps3-business-wp-257612.pdf.

[22] Oracle, “Oracle webcenter portal: framework” [Online]. Available: http://docs.oracle.com/cd/E23943_01/webcenter.1111/e10147/Topic_5.1. htm

[23] Oracle, “An introduction to oracle SQL developer data modeler,” [Online]. Available: http://www.oracle.com/technetwork/developer-tools /datamodeler/sqldeveloperdatamodeleroverview-167687.html.

[24] Oracle, “Oracle SQL developer for database” [Online]. Available: http://www.oracle.com/technetwork/developer-tools/sql-developer/sqldeveloperwhitepaper-v151-130908.pdf.

[25] Oracle, “Oracle database 11g express edition” [Online]. Available: http://www.oracle.com/technetwork/products/express-edition/overview/index.html

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California